REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 9, 1998	

4. TITLE AND SUBTITLE
Software Package for Speaker Independent or Dependent Speech Recognition
Using Standard Objects for Phonetic Speech Recognition

5. FUNDING NUMBERS
N61339-96-C-0088-P00001

Steve Broemker

6. AUTHOR(S)

H. Pfister

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Standard Object Systems, Inc.
1229 Tunica Drive
Opelousas, LA 70507

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Air Systems Command
Naval Air Systems Command HQ
Washington, D. C. 20361-0001

Naval Air Warfare Center
Training Systems Division
Code 4913
12350 Research Parkway
Orlando, FL 32826-3275

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

ABSTRACT ***(Maximum 200 Words)***

The Standard Object Systems, Inc. (SOS) SBIR Phase I and Option period research identified the overall requirements of an end to end phonetic speech recognition process for an SOS speech recognition software product (SPSR). The effort has focused on expanding the existing SOS Standard Objects for Phonetic Speech Recognition technology to the design of innovative software for a speaker independent continuous speech recognition system that can exploit the current parallel digital signal processing and mircocomputer technology. This software product for speech recognition will be developed in Phase II for the Naval Air Warfare Center Training Systems Division at Orlando, FL for use in speech recognition for air traffic control training.

14. SUBJECT TERI

15. NUMBER OF PAGES
43

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Distribution Unlimited

DTIC SCIENTIFIC AND TECHNICAL REPORT

WA 98-05-1366 QA REPORT NUMBER

2 copies of the report are being forwarded.

DISTRIBUTION STATEMENTS: Check the appropriate distribution statements.

[X] A. APPROVED FOR PUBLIC RELAEASE - DISTRIBUTION IS UNLIMITED

[ ] B. DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY. (Provide reason and date):

OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO: (Indicate controlling DOD office):

[ ] C. DISTRIBUTION AUTHORIZED TO U.S. GOVERNEMENT AGENCIES AND THEIR CONTRACTORS. (Provide reason and date):

OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO: (Indicate controlling DOD office):

[ ] D. DISTRIBUTION AUTHORIZED TO DOD AND U.S. DOD CONTRACTORS ONLY. (Provide reason and date):

OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO: (Indicate controlling DOD office):

[ ] E. DISTRIBUTION AUTHORIZED TO DOD COMPONENTS ONLY. (Provide reason and date):

OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO: (Indicate controlling DOD office):

[ ] F. FURTHER DISSEMINATION ONLY AS DIRECTED BY: (Indicate controlling DOD office and Date) Or HIGHER DOD AUTHORITY.

[ ] X. DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DOD DIRECTIVE 5230.25. WITHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE, 6 Nov 1984 (Indicate date of determination).

CONTROLLING DOD OFFICE IS (Indicate controlling DOD office).

[ ] This document was previously forwarded to DTIC on ____________ (date) and the AD number is: ________________.

[ ] In accordance with provisions of DOD instructions, the document requested is not supplied because:

[ ] It will be published at a later date. (enter approximate date, if known).

[ ] Other: (Provide reason).

DOD Directive 5230.24 "Distribution Statements on Technical Documents", 18 Mar 87, contains seven distribution statements, as described briefly above. Technical documents must be assigned distribution statements.

Leora Jason	4/2/98	LEORA JASON	DSN 960-4305
Authorized Signature	Date	Print or Type Name	Telephone Number

SOS

Phase I Option Report

DoD SBIR 96.1 Topic N96-055

Software Package for Speaker Independent or Dependent Speech Recognition Using Standard Objects for Phonetic Speech Recognition

Prepared By

Standard Object Systems, Inc.
1229 Tunica Drive
Opelousas, LA 70507

For Contract

N61339-96-C-0088-P00001

on

February 9, 1998

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED.

Phase I Option Report - DoD SBIR 96.1 Topic N96-055

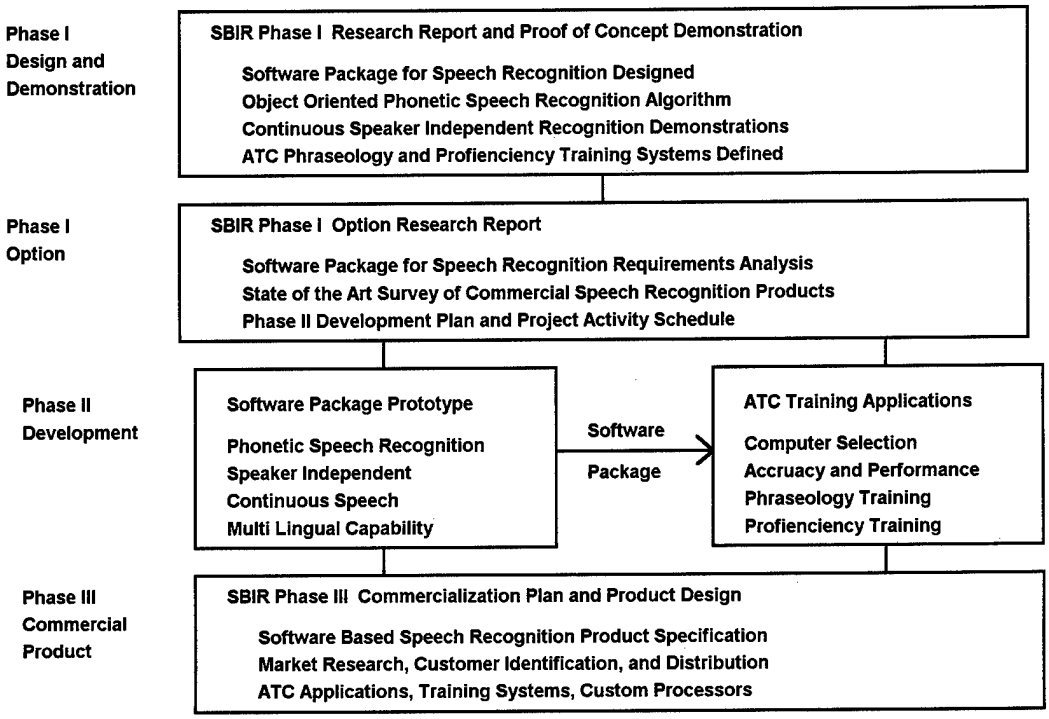
Software Package for Speaker Independent or Dependent Speech Recognition Using Standard Objects for Phonetic Speech Recognition

CONTENTS

1.0 PHASE I OPTION RESEARCH RESULTS

The Standard Object Systems, Inc. (SOS) SBIR Phase I and Option period research identified the overall requirements of an end to end phonetic speech recognition process for an SOS speech recognition software product (SPSR). The effort has focused on expanding the existing SOS Standard Objects for Phonetic Speech Recognition technology to the design of innovative software for a speaker independent continuous speech recognition system that can exploit the current parallel digital signal processing and microcomputer technology. This software product for speech recognition will be developed in Phase II for the Naval Air Warfare Center Training Systems Division at Orlando, FL for use in speech recognition for air traffic control training.

Figure 1.1-1 Software Package for Speech Recognition Project



1.1 Speech Recognition Concept

Spoken communication is not a simple process. Speech encompasses a wide variety of words, accents, tones, and speaker variations. Many words sound identical but do not have the same meaning, and often different meanings are attributed to the same word, depending on the speaker. Electronic speech recognition has been an elusive goal of researchers since the 1930's. Time Magazine illustrated the difficulty of this technology in the January 13, 1997 issue, where it stated that Bill Gates refers to his "voice recognition team as the 'wreck a nice beach' group, because that's what invariably appears on the screen when someone speaks the phrase 'recognize speech' into the system."

Most speech recognition approaches attempt to exploit numerous special circumstances to achieve reliable and rapid performance. In the end these circumstances, whether limited command vocabularies, discrete word pronunciation templates, various parameter quantizations, or complex finite state Markov models, all fail in the real world of noisy speech processing. The Standard

Object Systems, Inc. approach to speech recognition is a language based approach. It is a fundamental analysis of the basic unit of all spoken languages, the phoneme. The reliable detection, classification, and identification of a spoken phoneme is the key to a successful phonetic based speech recognition.

This novel SOS approach to speech recognition is based on digital signal processing to detect and identify phoneme signals rather than on language dependent processing models, acoustic templates of utterances, or statistical models of word dictionaries. Phonemes are the common speech segments produced by all speakers in all languages that remain relatively constant for short periods of time. They are created dynamically by a complex vocal tract filter applied to acoustic energy generated by pulses of air and radiated by the lips, nose, and cheeks. A speaker concatenates phonemes to produce words and phrases with unique personal accents and pronunciations.

The SOS approach analyzes the speech to detect a set of features that characterize the underlying phonemes for short time periods. A number of parallel classification algorithms analyze the features to estimate the actual phoneme for each short time period. The resulting estimates are statistically combined to determine the most likely phonemes in an utterance. Linguistic and lexical methods are used to convert the resulting phonetic segments to orthographic text for computer processing or machine instruction. These unique SOS algorithms accommodate most spoken languages, provide speaker independence, and are naturally continuous.

The applications for reliable speech recognition as a computer component are limitless. Speech is the next human machine interface paradigm for computer input. The microphone will augment the keyboard and the mouse as standard items on a desktop computer. Speech recognition is the natural solution for the bulky keyboard accommodation problem in portable computing. A touchpad and a microphone will eliminate the need for a traditional keyboard and free up the constraints of clamshell packaging. The estimate of the market size for speech recognition equipment is one billion dollars in 1998 as reported by several computer trade journals. This is a technology with intense international competition; a technology that presents a market advantage for the economy of the country that creates a usable commercial product.

During Phase I, an outline of the commercialization plan for a speech recognition software package was created. It will be expanded during Phase II to include phraseology and proficiency training prototypes as by-products of the speech recognition research. During the Phase I Option, SOS acquired three commercial speech recognition products and conducted an evaluation study of the current state of the art in commercially available speech recognition. Figure 1.1-1 summarizes the concept of the project.

Figure 1.2-1 Key Design Objectives for the Phase I Project

DESIGN OBJECTIVE PHASE I	APPROACH TO IMPLEMENT
Speaker Independent Recognition	Use training samples from TIMIT database
Real Time Response	Pentium with parallel digital signal processors
Continuous Speech	Limited by noise, processing, and intelligibility
No Vocabulary Limitation	SOS phonetic dictionary has 180,000+ entries
Multi Lingual Capability	IPA for 350 languages
Phrase Structure Grammar Definition	Syntactic parsing and semantic analysis
Punctuation, Numbers, Abbreviations	Lexical models to convert phonemes to text

1.2 Objectives of Phase I Research

In the Phase I report SOS defined a set of research objectives and success criteria for the Phase I project. SOS has found that, rather than subjective assessment, a formal analysis of our success is of great value in the development of our products and tools. The success criteria were not a minimum set of specification goals to satisfy the contract, but the set of research objectives to be accomplished in Phase I that will insure the success of the Phase II effort. These objectives are restated in Figure 1.2-1. During the Phase I Option period these objectives were reviewed.

The goal of Phase I and the Option period was the design of a true phonetic based, speaker independent, and continuous speech recognition software system. The Phase I research efforts culminated in a demonstration of voice recognition of the ATC vocabulary which includes international phonetic alphabet examples. A preview of an ATC Phraseology Trainer was added as an additional demonstration during Phase I.

Figure 1.2-2 Success Criteria for the Phase I and Option Research Effort

CRITERIA	RESULT
Evaluate the Software Package for Speech Recognition Requirements for Satisfaction by the Standard Objects for Phonetic Speech Recognition Process	Evaluation Successful
Demonstration of a Multi Lingual Phonetic Speech Recognition Capability for Real Time, Continuous Speech, Speaker Independent Phase II Prototype	ATC Demo & Grammar
Design of Software Based Speech Recognition that Exploits the Current Parallel Digital Signal Processing and Microcomputer System Technology	Design Complete
Phase I Option Requirements Analysis and State of the Art Survey	Complete

The success criteria for Phase I was a demonstration that the Phase II prototype software package is feasible as shown in Figure 1.2-2. A demonstration showing speaker independent, continuous real time voice recognition was the target result, with the obvious understanding that the demonstration would not perform to the extent of a product scheduled for completion after twenty four months of additional effort. The goal of Phase II is to achieve 95% recognition accuracy.

The SOS demonstration in Phase I is the proof of concept for the technical feasibility of Phase II speech recognition software package. The Phase I option concentrated on a requirements analysis and the planning for the Phase II project. SOS has completed the Phase I Option and is confident to proceed with the Phase II software development.

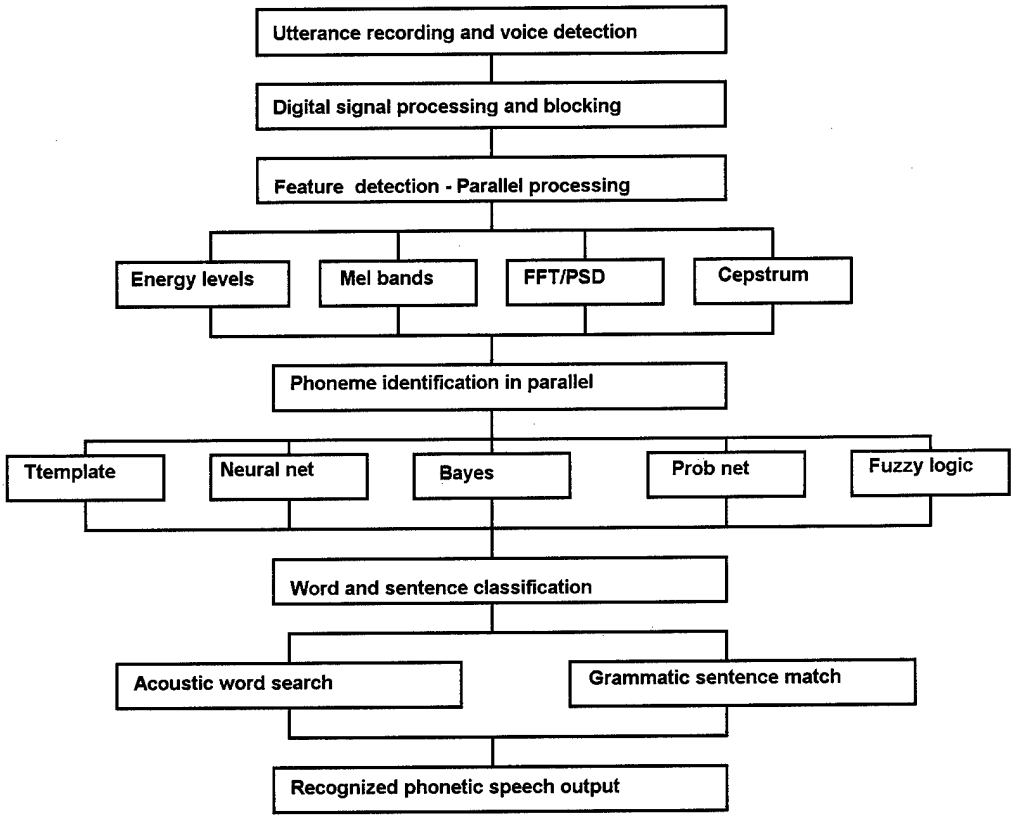
The Phase II software package is designed to be a fully functional prototype. It will support phonetic speech recognition in a number of applications on multiple computing platforms. The real time and parallel processing methods will be implemented to support parallel execution of DSP units. The prototype package will allow interactive execution, graphical debugging, and will support stand alone C++ object oriented applications in continuous speech recognition and speaker independent communication for air traffic control training. In addition, SOS will create a phonetic speech recognition tool kit as a commercial product.

1.3 Research Conducted During Phase I and the Option

The Phase I and option period research focused on two major areas. First, an analysis and definition of the requirements for a speech recognition software package were performed. Second, a design was created, and a proof of concept demonstration to test and evaluate the application of the existing SOS Standard Objects for Phonetic Speech Recognition was developed.

The nature of the research involved the design of an intricate network of algorithms which combine to produce the highest probability that a sound is recognized correctly as a given sequence of words. Details of the network design and the interconnectedness of the algorithms are provided in Section 2; the details of the mathematical computations are described in Section 5. Figure 1.3-1 summarizes the performance chain and the parallel intensive computational flow this research has produced.

Figure 1.3-1 Data Flow and Computation for Phonetic Speech Recognition Objects



In the figure, an utterance is recorded and the voice input is detected for speech recognition. Digital signal processing is performed to create short time data blocks for phonetic speech recognition. The speech features are detected for each data block with a set of parallel processing computations. These features are used for phoneme identification and are processed by a series of parallel classification algorithms to classify the voice features into phonetic units. The phonemes are statistically analyzed to match words and phrase segments with structured grammatical meanings. The recognized speech is output as text for further processing.

The benefits of the SOS Phase I and Option period research project are abundant. By using the Phase I results in Phase II, SOS is prepared to construct a fully functioning prototype software package for phonetic speech recognition. The prototype design will include the creation of application interfaces that allow Naval ATC facility users to evaluate the package in complex applications on multiple computing platforms and networks. User feedback will be reviewed to enhance and polish a commercial product in Phase III.

The results from Phase I are directly applicable to the research goals of Phase II. The Phase I proof of concept for speech recognition design will be used as the starting point for the Phase II prototype development. The Phase II goal is a software package prototype with a graphical user interface, generic phonetic speech recognition, air traffic control training application, multiple execution paradigms for parallel processing, real time digital signal processing, a network server interface, and a complete accomplishment of the requirements identified in Phase I.

1.4 Deliverables & Commercialization

The research and development will be directed in Phase II to the construction and testing of a fully functioning prototype software package for speech recognition, and in Phase III to the production of a commercial speech recognition software package product. The software package design incorporates the unique SOS phonetic speech recognition system that uses the international phonetic alphabet (IPA) as the smallest identified spoken unit. It supports both discrete word recognition and continuous speaker independent recognition with automatic training for a multi lingual speaker environment.

Figure 1.4-1 Deliverable Phase II Configurations for ATC Training Applications

MAC	CONFIGURATION	CAPABILITY
	Year I Alpha Unit	
6	**1. Design SPSR Software**	**Phonetic Speech Recognition Software Package**
12	**2. Phraseology Trainer**	**Evaluate the SPSR Design and Operation**
	Year II Beta Unit	
18	**3. Proficiency Trainer**	**Analyze 15G33 Proficiency Trainer to use SPSR**
24	**4. Final Phase II SPSR**	**Test with NAWC TSD Air Traffic Control Speech**
30	**Phase II Option**	
	5. Demo of SPSR with 15G33	**Apply SPSR Tool Kit to the Proficiency Trainer**

Figure 1.4-1 illustrates the configurations that will be developed by SOS during Phase II. In the first year, SOS will deliver an alpha prototype to demonstrate ATC phraseology training on a low cost platform. In the first year configuration, the phraseology trainer will prompt a student with an ATC phrase and evaluate the response. This will improve student performance of ATC speech requirements. The phraseology trainer will also perform phonetic segmentation of the student response to detect words. Through this process a database of words is collected so the student enrollment session can be replaced with phraseology training.

In the second year of Phase II, SOS will develop a beta prototype, a fully functioning ATC speech recognition unit applicable to phraseology training. The third configuration will analyze the stand

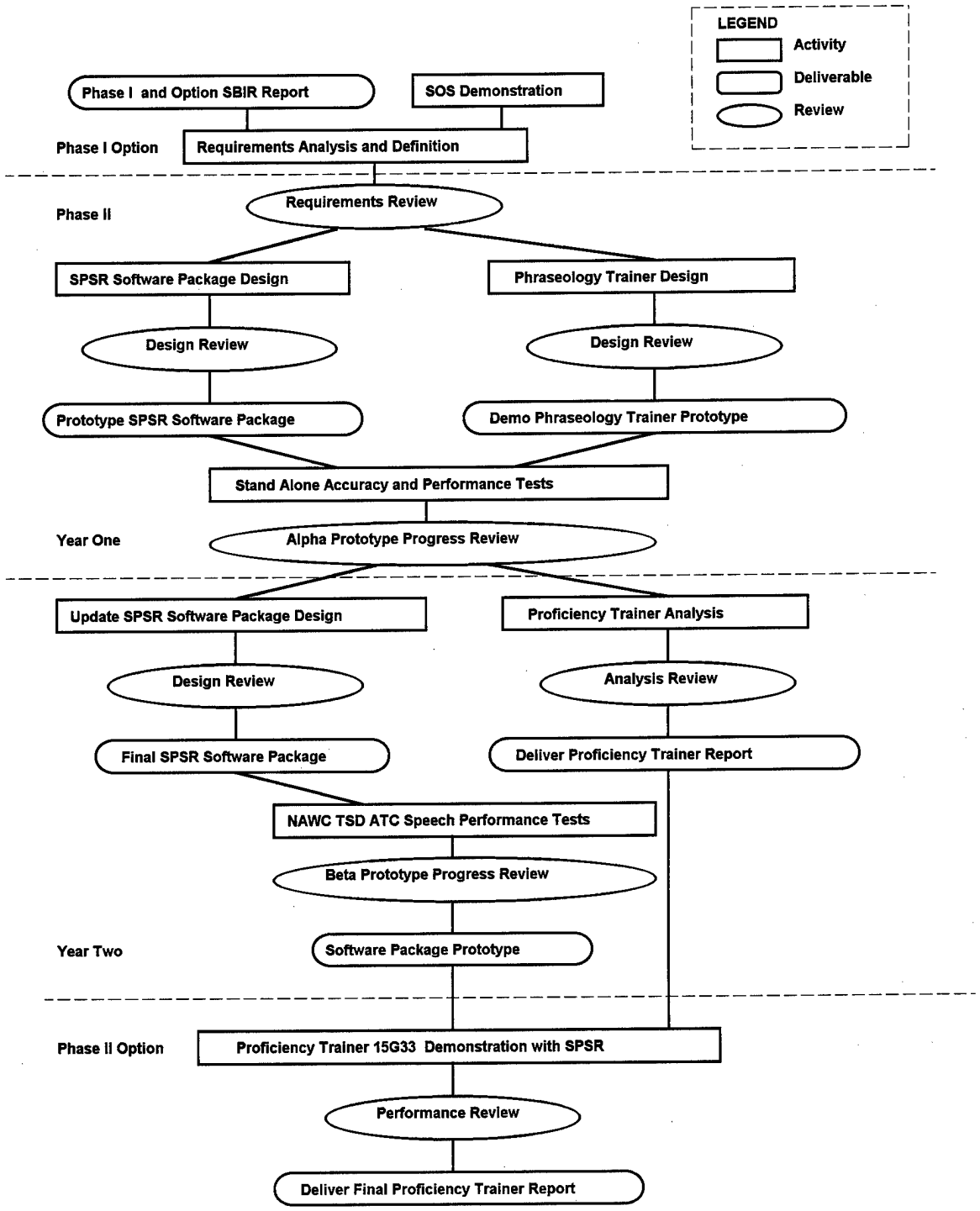
alone 15G33 proficiency trainer for accuracy and performance testing. The final configuration will be the SPSR software package integrated into the ATC training environment for operational evaluation with NAWC TSD speech inputs. At the end of Phase II the prototype SPSR will contain the operationally tested and evaluated speech recognition software package as the final product of Phase II. The road map for the Phase II project and Option period is presented in Figure 1.4-2.

Both the alpha and beta SPSR prototypes developed during years one and two of Phase II will stand alone as marketable speech recognition tool kit products. The phraseology trainer (alpha) will be in demand anywhere individuals are taught to communicate in a restricted phrase pattern and are required to be clearly understood. Police dispatchers and 911 emergency operators are obvious examples of situations where a student phraseology trainer could provide a dramatic increase in performance in a short amount of time. As the phraseology trainer provides each student feedback on how clearly he speaks, the trainer collects statistical data on the student performance which provides evaluation resources for the instructors.

The SPSR tool kit demonstrated in the Phase II option will build on the alpha and beta units by adding features suitable for commercial speech recognition applications. For the Navy, it can provide continuing professional education for certified air traffic control personnel. It could also provide training in ground control protocols for flight school attendees and students learning radio communication. The SPSR software package, which drives both the alpha and beta prototype deliverables, will have been fully tested and be operational at the end of Phase II for use in the Phase II Option demonstration with the 15G33 proficiency trainer. The SPSR will be ready for Phase III commercial marketing as Version 1.0. It is this version which will be available for integration into a future Naval ATC training system architecture as the SOS speech recognition component.

The key advantage to using SOS Standard Objects for Phonetic Speech Recognition is that the same software will operate over a wide range, from a discrete word recognition mode to a continuous speaker independent mode. SOS C++ objects are both scaleable in the use of computing time and modular in processing structure to adapt to a wide range of parallel and distributed heterogeneous computing environments. The SOS algorithm can be adjusted so that the more computer processing resources available, the higher the recognition accuracy. It can be extended to include the results of future research and development into both the identification of additional phonetic features and the development of specialized classification methods. The recognition processes are designed to operate in either a real time or a latent mode, depending on the platform and operating system resources. Because of the fundamental phonetic basis of the SOS approach, it provides a natural interface for multilingual lexical, syntactic, and semantic language processing systems. This process is unique by not having any inherent language assumptions that will limit its performance in a multi lingual environment.

Figure 1.4-2 Roadmap of Phase II and Option for the SPSR Software Package



2.0 PHASE I TECHNICAL OBJECTIVES

The Phase I focus is on the design of the proposed SPSR technology, a speaker independent or dependent, real time voice recognition software package to be used in the Naval Air Traffic Control training school as shown in Figure 2.0-1. The first step in the design was the determination of a list of required attributes of the product. Section 2.1 outlines these required attributes for the speech recognition ATC training software. Section 2.2 gives a short overview of

Figure 2.0-1 Software Package for Speech Recognition Design and Development

DESIGN PHASE	ACTIVITIES
Software Only Development in C++	Continuous Speech Recognition Algorithm TIMIT Speaker Database Interface and Training Phonetic Dictionary Interface Phrase Structure Command Grammar Non Real Time Verification and Validation Hardware Selection for Deliverable Unit Algorithm Documentation
Phonetic Feature DSP Speech Input	Live Speaker Input Test Speaker Independent Demonstration Non Real Time Recognition Human Computer Interface Performance Documentation
Parallel Phoneme Detection DSP	Real Time Recognition Test Network Communication Protocol Interface Parallel DSP Execution Demonstration Performance Analysis and Noise Testing Final Prototype Hardware Configuration Operational Documentation
Integration with Training Operations	Spoken Command and Control Test User Interface Demonstration Operational Software Interface Commercial Prototype Definition Production Hardware Specification System Delivery Documentation

object oriented programming and relates how the SOS Standard Objects for Phonetic Speech Recognition method painstakingly addresses the requirements already listed in Section 2.1. Section 2.3 will discuss a number of performance criteria for speech recognition. Before hardware specifications are considered in 2.5, real time parallel processing and its computing power requirements will be discussed in 2.4.

2.1 Phase I Software Package Requirements Analysis

The SPSR software is structured to encompass the technical objectives and design elements listed in Figure 2.1-1. The technical objectives are: speaker independent operation, continuous speech processing, phonetic speech recognition, large vocabulary capacity, multi lingual operation, phrase structured grammars, operational command recognition, real time response, parallel digital signal

processing, and local area network compatibility. The following sections summarize the Phase I technical objectives which will be the basis of the Phase II requirements specification.

Figure 2.1-1 Software Package Phase I Technical Objectives and Methods

OBJECTIVE	METHOD
Requirements Analysis	Collect and analyze requirements for speech recognition topics: Speaker Independent, Continuous Speech, Natural Language, Multi Lingual, Real Time Performance, Parallel Processing, Local Area Network Access, Add On Module
Software Package Design	Synthesize a software package to satisfy the independent and dependent speech recognition requirements and addresses the design and execution issues. Document and review design.
Speech Recognition Process	Design a speech recognition prototype in Phase II based on Demo of Hiragana prototype in Phase I that satisfies the software package design and performance requirements.
Software Design	Apply object oriented programming methods using C++ to the Phase II software package, the DSP environment, the speech recognition application, and the integrated execution.
Hardware Specification	Specify the hardware to support the Phase II software package development, stand alone testing, training environment, Integration on a network, and operational testing.
Integration and Testing	Plan the integration and test of the software that includes Digital Signal Processing for Continuous Speech Recognition.

2.1.1 Speaker Independent and Dependent Operation

A new student in the Air Traffic Control Training Center presently spends time at a computer terminal familiarizing the computer with the various speaking idiosyncrasies of his speaking voice. For the new enrollee and the training staff, this process can be torturously frustrating and time consuming. Speaker independent voice recognition implies a software package design that can be operated without first training the system to each individual speaker.

A more beneficial training is for the student to be lead, through the equipment, to better speak the various commands and ATC language vocabulary before he is exposed to the actual training apparatus. This familiarizes the student with equipment interface, and gives positive direction in learning to speak the commands comfortably and with confidence. Although the SOS software will not require training as before, the performance of the software will be maximized if the students speak in such a way they know the computer can comprehend their commands. Thus a student phraseology trainer demonstration is proposed during Phase II.

2.1.2 Continuous Speech Processing

The requirement for continuous speech processing implies a software package design that accepts natural spoken utterances and does not require artificial word isolation or pronunciation. While the ATC has a defined vocabulary, it is impracticable not to expect variable dialogue in crisis

situations when clear communication is most necessary. A variety of response is inherent in any emergency situation.

The SOS approach for the continuous speech requirement is to use speech recognition for the detection of each phonetic utterance, the phoneme. The utterance is defined as a continuous sound between silent periods of at least a given duration. The SOS phonetic speech recognition is designed specifically to deal with varying speaking rates, coarticulation of phonemes, and non speech sound artifacts. While a ATC vocabulary will be factored in as most-used phrases, the software should have full lexical recognition to be realistically viable and commercializable.

2.1.3 Phonetic Speech Recognition

The requirement for phonetic speech recognition implies a software package design compatible with an existing phonetic alphabet. To satisfy this requirement the SOS approach uses phonemes that are applicable to 350 spoken languages as the basic units of speech recognition. No additional phoneme data should be needed for this specific speech recognition software package application. However, the phonetic set can be extended easily with additional phoneme definitions for a wider range of performance.

Various linguistic sources have reported that English can be represented by as few as 32 phonemes to as many as the 62 phonemes as used in TIMIT. Figure 2.1.3-1 illustrates the SOS definition process for the phonetic alphabet. In the Phase I study, the initial set of 62 phonetic symbols was used as the baseline alphabet. A count was done of the frequency of occurrence of these symbols in the TIMIT vocabulary of 6224 words. Symbols that had a zero occurrence in the TIMIT data were eliminated for a reduction of 17 symbols. Symbols that had a small frequency of occurrence or that were phonetically indistinguishable were combined for a reduction of 9 additional symbols resulting in the 36 symbol alphabet.

2.1.4 Large Vocabulary Capacity

The requirement for a large vocabulary capacity implies a software package design that is phoneme oriented and compatible with existing phonetic dictionaries. A phonetic dictionary is a collection of sound combinations which have word meanings. The phonetic dictionary categorizes by phonetic spelling. For example, the sound TU could have any of the meanings lexically referred to as to, too, or two. In speech recognition the software must determine from the context of the surrounding utterances which meaning is appropriate.

The SOS approach for a large vocabulary capacity requirement is to collect a multi lingual phonetic dictionary with approximately 180,000 word/sound entries. Additional phonetic text data has been collected for the specific Air Traffic Control application which will extend the dictionary for a wider range of performance. The CATCC example used in the Phase I study demonstration had a vocabulary of over 400 words. The TIMIT training database has a vocabulary of over 6000 words.

Figure 2.1.3-1 Selection and Definition of the Phonetic Alphabet

AR - TIMIT/Arpabet (62 symbols)
PA - Phonetic Alphabet Symbols (1-36)
CNT - TIMIT Frequency of Symbol in Vocabulary
I Index (0-61) of Symbols in TIMIT

AR	CNT	PA	I	AR	CNT	PA	I	AR	CNT	PA	I
b	723	22	0	h#	0	0	20	el(l)	410	3	41
d	1691	20	1	bcl	0	0	21	iy	1302	17	42
g	453	25	2	dcl	0	0	22	ih	1689	18	43
p	1162	23	3	gcl	0	0	23	eh	992	14	44
t	2468	21	4	pcl	0	0	24	ey	709	19	45
k	1645	26	5	tck	0	0	25	ae	952	15	46
dx	0	0	6	kcl	0	0	26	aa	630	11	47
q	0	0	7	tcl	0	0	27	aw(ao)	163	12	48
jh	300	31	8	m	1086	8	28	ay(ey)	623	19	49
ch	212	32	9	n	2164	9	29	ah	531	13	50
s	2333	35	10	ng	509	10	30	ao	489	12	51
sh	484	33	11	em(m)	29	8	31	oy(iy)	83	17	52
z	1223	36	12	en(n)	95	9	32	ow	523	4	53
zh	41	34	13	eng	0	0	33	uh(ao)	190	12	54
f	689	28	14	nx	0	0	34	uw	441	2	55
th	153	30	15	l	1695	3	35	ux	0	0	56
v	559	27	16	r	1900	7	36	er	314	6	57
dh	89	29	17	w	395	5	37	ax	1301	1	58
pau	0	0	18	y	253	16	38	ix(ih)	1517	18	59
epi	0	0	19	hh	325	24	39	axr(er)	862	6	60
				hv	0	0	40	ax-h	0	0	61

2.1.5 Multi Lingual Capability

The Naval ATC program directors have not required the software to perform speech recognition in any language other than English. The commercialization prospects of a final product, however, are enhanced if it can be marketed for a wide range of languages. This can be accomplished with minimal additional expense. Therefor, the package design will be compatible with the existing international phonetic alphabet for multiple languages and dialects. The SOS approach for this requirement is to use an existing IPA which applies to the most common 350 spoken languages. No additional phoneme data is anticipated for this specific speech recognition application. However, the IPA may be extended easily for a wider range of phonetic languages or dialects.

2.1.6 Phrase Structure Grammar

The requirement for phrase structure grammar recognition implies a software package design that is compatible with natural human communication. The SOS approach for this requirement is to use a simple phrase grammar which applies to ATC voice command recognition and equipment control/response applications. Additional grammatical data may be added for a wider range of applications such as text transcription.

2.1.7 Air Traffic Control Training

The requirement for air traffic control training implies a software package design compatible with a human machine interface which is highly reliable for air traffic control training operation. The SOS approach for this requirement is to use ATC defined phrases which apply to the training applications, including positive

verification of command sequences. Numerous command phrases may be defined automatically using the phonetic dictionary for a wide range of training applications.

2.1.8 Real Time Response

The requirement for real time response implies a software package design that performs within the normal human communication response time. The SOS approach for this requirement is to use a phonetic speech recognition algorithm that is computed on parallel digital signal processors to achieve real time performance. The algorithm is composed of computationally predictable phases that are not susceptible to software logic failures or numerical convergence inherent in other recognition methods. The phonetic speech recognition process is scaleable, so that more computing resources will produce more accurate results. The process, however, may be terminated at any time to produce the best estimate of the spoken input for critical response systems.

2.1.9 Parallel Digital Signal Processing

The requirement for parallel digital signal processing implies a software package design partitioned into independent and concurrent computational phases. The SOS approach for this requirement is to use its unique phonetic speech recognition algorithm which is defined as a sequence of linear communicating processes that adapts easily to a network of heterogeneous parallel processors. The construction of this software is such that it will compute on two parallel paths 1) the analysis of the features of spoken utterances and 2) the likelihood that the phonemes within the utterance compose a given string of words. Each path requires algorithmic permutations that are performed both in sequence and in parallel on each path, to be merged late in the processing stage.

The algorithm is composed of computational phases that can be either data parallel or process parallel depending on the selected computing architecture. SOS has defined a unique method for structuring parallel processing using multiple execution paradigms that are defined in the application. This differs from most approaches which depend upon automatic parallelization of a complex computer program, either at compile or execution time.

2.1.10 Local Area Network

The requirement for local area network implementation implies a software package design that can interface and communicate using standard protocols. The SOS approach for this requirement is to use a network interface layer compatible with TCP/IP, ATM, and many other standard protocols for external communications with the Software Package. The phonetic speech recognition algorithm uses its own efficient computational communication protocol for high performance parallel internal computer communication.

2.2 Object Oriented Software Development

Computer programming for procedural computations has developed in four distinct phases as shown in Figure 2.2-1. Programs were designed first with small pieces of working machine code which was tested and layered to form larger programs. This was successful for small projects but failed on larger projects due to the chaotic nature of the interfaces. The second phase was procedural programming using languages such as FORTRAN and C. Computational functions were embedded in procedures and called from other

procedures. This is still the most common development process, but has weaknesses in reuse of code and in support of data oriented problem analysis.

The third phase saw database oriented fourth generation languages that automatically created large amounts of correct procedural code. The major weakness is the limited analysis domain and the difficulty of integrating existing programs. The fourth phase is the use of object oriented methods that combine data and procedures into abstract data types called objects. Objects send and receive messages that define their behavior. The four major characteristics of objects are encapsulation of data and computation into a single entity, abstraction to allow objects to exhibit expected behavior from standard operators, inheritance of behavior by new objects for code reuse, and polymorphism to redefine inherited functions as needed.

Figure 2.2-1 Four Phases of Procedural Programming Languages

PHASE	DESCRIPTION
Machine	Small segments of working code layered to form a program
Procedural	Computations are contained in procedures and functions
Data	Data oriented computations and procedure generators
Object	Combine data and procedure to form objects and messages

2.2.1 Object Oriented Design

Object oriented software design creates systems as dynamic networks of interacting objects. Each object is responsible for carrying out functions requested from other objects using messages. The overall design goal is to define an independent set of objects that exhibit a high degree of internal cohesion and are loosely coupled to each other. A good design retains a close mapping between the problem domain and the computational solution. The steps in designing an object oriented system are given in Figure 2.2.1-1. Each of the object oriented design steps is difficult to do well. A number of techniques exist to aid in the design process. Some use complex notation, scenario analysis, or scripts. In the end, the design is a set of objects and methods structured by inheritance, composition, association, and behavior.

Figure 2.2.1-1 Object Oriented Software Design Process

Identify the objects in the problem domain
Construct the relations between objects
Select the object names and messages
Structure the objects to delegate system capability
Factor the objects for ease of reuse

2.2.2 Object Oriented Programming in C++

The language C++ was originally developed to write event driven simulations. It used the object concepts defined in Simula67, operator overloading from Algol68, and could be directly translated into C for compilation and execution. Since C++ retains C as a proper subset, it is easily learned by existing C programmers, and legacy C code can be encapsulated into C++ objects. These are both advantages, from a developers view, and disadvantages from the view of the language designers.

Encapsulation in C++ is the combining of data types and member functions into a class which defines an object. Data types can be types or objects while member functions have an identical syntax to ordinary C functions. Classes do not allocate storage, only object instances of classes allocate storage. Abstraction in C++ requires that a new class specify class behavior for operators by overloading them. Inheritance in C++ defines a new child class that inherits the characteristics of the parent class. Polymorphism is implemented by overriding the member function definitions of the parent in a child class. Member functions can be overridden with virtual functions that are pointers to functions.

C++ is a complete object oriented language and forms the basis of the textual programs generated from object oriented language computational models. Designers use visual blocks to create instances of domain classes. Blocks accept blocks as messages and produce blocks as messages. The links between blocks define the routing of the object messages. In addition, a visual object design language can be written in C++ so that it is portable along with the application programs.

Figure 2.2.3-1 Object Design Patterns

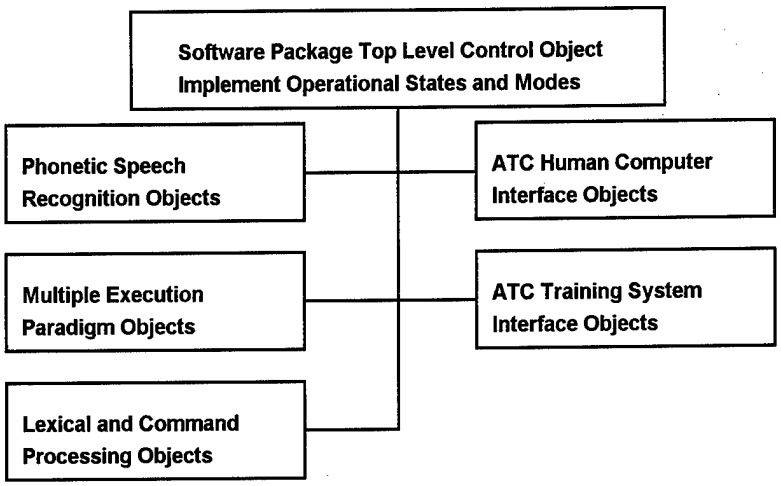
CREATION	STRUCTURE	BEHAVIOR	CONTROL
abstract class	adapt class	chain requests	iterate objects
build object	decouple abstraction	encapsulate request	capture state
create method	compose objects	interpret grammar	state change
copy object	attach object	mediate interaction	algorithm selector
single instance	unified interface	dependent objects	skeleton algorithm
	share objects	add operation	
	surrogate object		

2.2.3 Standard Object Programming Methodology

SOS has a methodology for object oriented design and programming that is suitable for use on a moderate size project involving a small team of programmers. It consists of guidelines for object oriented design, class definition, file organization, naming, coding, and documentation. The purpose is to produce C++ programs that are consistent and reusable. The design of modular software is not a new or novel approach. However, the SOS use of software tools to construct objects that meet rigid standards for logic and data, and which interface so that complex systems can be generated directly from system requirements and easily verified, is a novel and innovative approach to software development.

Designing reusable objects is hard. Given the application requirements, you must identify objects, combine them into classes, define interfaces, create hierarchies, identify associations, and prepare for parallel computation Good object designs have recurring patterns for classes and communications that provide flexibility and reusability. Object design patterns provide a standard solution to common problems that can be easily reused. These generally consist of communications objects and classes that can solve general design problems in specific contexts. SOS has created a set of design patterns shown in Figure 2.2.3-1. These patterns are divided into four groups. The creation class defers some part of an object definition to subclasses and objects to another object. The structure class uses inheritance to compose classes while structure objects define ways to assemble objects. The behavior groups define operations for groups of classes. The control group provide structures for objects and states.

Figure 2.2.4-1 Software Package Top Level Control Object



Class design definitions must contain the following items. The external design of the class is the interface, which is contained in the header file and consists of the member function definitions. The internal design includes the class data elements, classes included by composition, and internal functions. The inheritance structure of the objects defines a "kind of" relationship between objects. The composition relationship or "part of" definition specifies the parts contained in an object. The file organization is usually a header and a source file for each class. The naming convention is usually referred to as the polish form. Coding conventions stress simple code to promote understanding, readability, and maintainability, which, as a general rule, provide one statement per line with explanatory comments. SOS documentation is of commercial quality similar to most modern technical and user manuals.

2.2.4 Software Package for Speech Recognition Objects

The software package will consist of six software object groups. A top level control object directs the computations of the lower level object groups as shown in Figure 2.2.4-1. The lower level object groups are divided into two separate categories for speech recognition and system interface that correspond to the alpha and beta configuration developments and delivery schedule. The software will be developed primarily in C++ using the SOS object oriented software development approach, however, it may be necessary to utilize assembly code for small portions of the DSP software to interface efficiently for optimum performance.

The speech recognition object group contains the phonetic speech recognition objects, the multiple execution paradigm objects, and the lexical processing objects; all designed using SOS Standard Objects for Phonetic Speech Recognition. The phonetic speech recognition objects will implement the algorithms for phoneme feature detection and classification, Figure 2.2.4-2. The multiple execution paradigm objects in Figure 2.2.4-3 implement the parallel digital signal processing for the specific software package hardware configuration.

Figure 2.2.4-2 Phonetic Speech Recognition Object Group

OBJECT	BEHAVIOR
wave	Control digitization process, platform specific, real time
tilt	Pre emphasis spectrum tilt filter
block	Organize sound samples into segments for feature processing
feature	Control multiple parallel feature computations
energy	Compute segment energy and zero crossing features
mel	Compute Mel band FIR filter features
psd	Compute FFT and PSD features
cepstrum	Compute LPC and cepstrum features
classify	Control multiple concurrent classification algorithms
metric	Classify by minimum metric difference to templates
sample	Create sample templates for metrics
network	Classify by feedforward neural network
train	Compute weights for layers of network
bayes	Classify by Bayesian conditional probability statistics
covar	Compute feature covariance matrix and inverse
probnet	Classify by probabilistic neural network
stats	Compute statistics for probabilistic network
fuzzy	Classify by fuzzy logic and set computations
member	Compute membership functions for phonetic data
model	Create model data for classification algorithms
confuse	Compute confusion matrix to assess recognition performance
estimate	Combine classification probabilities for best phoneme estimate
dynamic	Discrete state dynamic programming to select best phonemes

The lexical processing objects in Figure 2.2.4-4 implement the phrase structure grammar for the Software Package air traffic control training applications. The interface object groups shown in Figure 2.2.4-5 are for the Beta configuration and contain the human computer interaction objects and the local area network interaction objects. The detail design of these objects is based on the requirements analysis of the Software Package operational environment. They will be developed for the Beta configuration, but they will have a C++ header specification defined for compatibility with the Alpha configuration.

Figure 2.2.4-3 Multiple Execution Paradigm Object Group

OBJECT	BEHAVIOR
process	Abstract processor model
message	Message passing communication model
share	Shared memory access model
task	Parallel task launch and control
status	Parallel real time task status information model
monitor	Access task in step mode for diagnosis
step	Incremental execution of a task for debug
stop	Halt the execution of any task or process
start	Initiate the execution of any task or process

Figure 2.2.4-4 Lexical Processing Object Group

OBJECT	BEHAVIOR
phonetic	Phonetic dictionary search by morph and syllable for utterance text
lexicon	Interface to phonetic dictionary for expansion and maintenance
syllable	Syllable recognition from phoneme string
word	Word recognition from syllable and phoneme data
phrase	Phrase recognition from words and syllables
parse	Syntactic and semantic parse of utterance for phrases
grammar	Phrase structure grammar rules and interpretation
syntax	Application command and control syntax specification
semantic	Semantic information network model for application
lexical	Spell, punctuate, and export text phrase
dialog	Input and response state machine for commands
reply	Question and information response generation
request	Operator question generation for additional information

2.2.5 Standard Object Cost Model (SOCOMO)

At the time that the SOS principal investigator was a member of the TRW technical staff, Ray Wolverton and Barry Boehm, also from TRW, developed a model for estimating the number of lines of code and the number of man hours required for a large software development project. This model, called COCOMO, was modified for Windows and object oriented programming by William Roetzheim. SOS expanded the cost projection model to include its standard object development approach, and refers to it as SOCOMO.

Figure 2.2.4-5 ATC Training Interface Object Group

ATC HUMAN COMPUTER INTERACTION OBJECTS

OBJECT	BEHAVIOR
type	Text input interface
point	Mouse input interface
click	Button input interface
speak	Speech input interface
display	Graphic output interface
sound	Audio output interface

LOCAL AREA NETWORK INTERACTION OBJECTS

OBJECT	BEHAVIOR
node	Network node interface
protocol	Communication protocol interface
send	Message transmission interface
get	Message receipt interface
manage	Network interface management control

The SOCOMO model is implemented on a spreadsheet to provide flexible experimentation of the parameters. There are four data sections in the model. First, the SOCOMO WINDOWS DATA that defines the basic model parameters. Second, the SOS experience data estimated from similar software developments. Third, the PHASE II DATA estimated for this specific software development program. Fourth, the SOCOMO results divided into two parts for the baseline object estimates and for the estimates

adjusted to compensate for the efficiency of the development process. In addition, two parametric computations are made for a low estimate and a high estimate. In general, this range will include the actual project results 68% of the time. However, as in all estimation models, the results are only as good as the quality of the input data.

The SOCOMO WINDOWS DATA items consist of the base ratio and nonlinear effect which estimate the months effort per thousand lines of code and a factor to compensate for the nonlinear nature of software development. The project ratio and the project effect convert months of effort into project months. The remaining parameters estimate the lines of code per object and the objects per program feature. The last three parameters convert lines of code into estimates of documentation pages for prototype commercial products.

For this project, the SOCOMO inputs and results are presented in Figure 2.2.5-1. The result of the model for the Phase II prototype Software Package development are the median estimates for total lines of C++ code (34,500), for the total man months of effort (99), for the project length in months (19.5), and for the number of commercial level documentation pages (240). This indicates a high probability that the project can be successfully completed within the time and budget of the two year Phase II SBIR program.

2.3 Speech Recognition Performance

The accuracy of phonetic speech recognition depends upon the correct and consistent identification of phonemes. This is like trying to detect and identify each individual word spoken by a Frenchman if you have only read French. To the untrained ear it is very difficult to identify single words in a language spoken normally. The SOS speech recognition approach requires even more challenge; it isolates each phonetic sound which may or may not have individual meaning. Then it computes the mathematical probability of the meaning of these sounds in combination. The SOS approach is also unique in that it is scaleable with the amount of computing performed. There are no arbitrary limits imposed by quantization, finite states, or dictionary size.

The goal for the implementation of the Japanese Hiragana syllable recognition was to correctly identify each syllable 90% of the time and to include the correct syllable in the top three choices 95% of the time. Similar targets apply for the ATC training and for other languages. The performance in terms of response time and memory requirement of the phonetic speech recognition is dependent on a number of controllable factors, see Figure 2.3-1. Specific performance is dependent on the selected hardware configuration.

FIGURE 2.2.5-1 SOCOMO - Standard Cost Model for Software Package Development

PROJECT: PHASE II LANGUAGE BASED SPEECH RECOGNITION DEVELOPMENT

MODEL	VALUE	SOCOMO / WINDOWS DATA
BASE RATIO	3.50	
NONLINEAR EFFECT	0.95	
PROJECT RATIO	4.50	
PROJECT EFFECT	0.35	
LINES PER OBJECT	15.00	
OBJECTS PER MENU	0.50	
OBJECTS PER WINDOW	5.00	
OBJECTS PER DIALOG	1.50	
OBJECTS PER CLASS	2.00	
OBJECTS PER MEMBER	14.00	
REQUIREMENTS PAGES RATIO	1.50	
DESIGN PAGES RATIO	3.50	
SPECIFICATION PAGES RATIO	2.00	

PROCESS FACTORS	SOS EXPERIENCE DATA
CAPABILITY	0.90
EXPERIENCE	0.90
COMPLEXITY	1.00
DIFFICULTY	1.10
SIZE	1.00
SCHEDULE	1.00
TOOLS	0.90
COMBINED FACTOR	0.80

PROJECT FACTORS	PHASE II SOFTWARE DATA
MENUS	50.00
WINDOWS	3.00
DIALOGS	5.00
FILES	2.00
CLASSES	20.00
MEMBERS	200.00

SOCOMO ESTIMATES	MEDIAN	LOW	HIGH
OBJECT UNITS	2,887.50	2,454.38	3,320.63
BASE OBJECT KLOC	43.00	36.00	49.00
BASE MAN MONTHS	124.00	105.00	141.00
BASE PROJECT TIME	24.32	22.94	25.44
BASE REQUIREMENTS	64.50	54.00	73.50
BASE DESIGN	150.50	126.00	171.50
BASE SPECIFICATION	86.00	72.00	98.00
ESTIMATED CODE (1000)	34.48	28.87	39.29
ESTIMATED MONTHS	99.44	84.20	113.07
ESTIMATED PROJECT	19.50	18.40	20.40
REQUIREMENTS PAGES	51.72	43.30	58.94
DESIGN PAGES	120.69	101.04	137.53
SPECIFICATION PAGES	68.96	57.74	78.59

Figure 2.3-1 Key Factors That Affect Phoneme Recognition

PROCESS	FACTOR	EFFECT
Sound Signal Processing	Sample A/D Rate	Segment Data Rates
	A/D Resolution	Feature Accuracy, Memory Size
	Segment Size	Latency, Phoneme Resolution
Phonetic Feature Generation	Silence Threshold	Utterance Onset or Termination
	Mel Filter Stages	Bandpass Rolloff, Time
	FFT length	Frequency Resolution
	Correlation Period	LPC Accuracy
Phonetic Detection	Number of templates	Phonetic Accuracy
	Neural Net Hidden Nodes	Classification Accuracy
	Sample size	Bayesian Accuracy
	Pattern Layer Size	Probabilistic Network Accuracy
	Membership Function	Accuracy of Fuzzy Classification
Phoneme Identification	Estimate Combination	Probability of Identification
	Lagrange Multiplier	Dynamic Programming Accuracy

2.3.1 Accuracy of Phoneme Identification

A key issue in speech recognition is how accurately phonemes must be detected to provide the best data for word recognition. An experiment was performed using digit recognition in English with multiple speakers to analyze the effect of phonetic endpoint errors on recognition accuracy. Recorded speech was digitized and examined to locate the phoneme endpoints by hand. This data was used as a reference set to build digit templates for recognition.

The recognizer was tested and performed at a 93% accuracy with the original templates. Phoneme endpoints were then varied in time by steps of 15 ms to test the accuracy. A 3% reduction of accuracy occurred with a 60 ms endpoint shift growing to a 30% reduction at a 120 ms shift. These results indicate that accurate location of phoneme boundaries is essential for accurate detection, which can only be achieved by the use of precise signal processing techniques.

2.3.2 Reliability of Speech Recognition

The reliability of phoneme recognition is the probability that the correct phoneme will be identified from the input speech data. It is usually portrayed using a confusion matrix where the rows are the spoken phoneme and the columns are the recognized phonemes. The matrix entries are the fraction of classifications of a known phoneme test set. The ideal performance would be a diagonal matrix of ones indicating perfect recognition. Off diagonal elements indicate confusion between phonemes, and can be used to tune or modify the algorithms. In general, an excellent performance is a 90% recognition. This translates into a much higher speech recognition index since a sample rate of one hundred per second will result in from five to over twenty samples per phoneme. Assuming the samples are independent and the noise is uncorrelated, the speech recognition rate will be 99%.

2.3.3 Acoustic and Electrical Noise

For speech produced in a noisy environment the accurate detection of phonemes is a complex task. Two classes of problems contribute to background noise. One class of noise problems is the ambient noise environment. Machines create noise. Even a recorder prints its own distortion signature on the sounds it copies, creating variations which blend and interfere with the pure tones of the speaker. Ambient noise is controlled by using a noise limiting microphone mounted near the speakers mouth. In addition, a running average of the background noise during periods of silence will be maintained to aid in the removal of noise from the signal. Sudden non vocal tract background noises, such as bells or machinery, are the second type of noise problem. These are easily classified and result in requests to repeat if the power level interferes with the speech signal.

2.3.4 Distortion and Speaker Noise

For speech produced in a distorted environment, the accurate detection of phonemes is more difficult. Two classes of problems contribute to this distortion. The first class of problems stem from the environment, such as echoes, and also from apparatus, such as the headset in aircraft pilot helmets. These sounds are removed by filtering. The other class of problems is attributed to the speaker who produces sound artifacts during speech, such as lip smacks, heavy breathing, mouth clicks, and nasal pops. Such artifacts are generated inadvertently, but are at an energy level comparable to speech. Editing of recorded speech to remove these artifacts has improved recognition and reduced confusion. In order to achieve a high level of performance, an adaptive recognizer will explicitly model speaker artifacts and detect them along with phoneme recognition.

2.4 Real Time Parallel Processing

Computer designers have long claimed that parallel processing can dramatically increase software application performance. Parallel processing is a method where the computational requirements of a software program are shared between two or more computers. The time required to perform necessary calculations is divided by the number of machines. With parallel processing a control program coordinates the division of processes and merges the processed data from the various machines back to the main terminal. Each machine added to the parallel processes shortens the computing time, but the control program in turn becomes more complex and more time intensive. For example, a program requires 12 minutes to run a set of computations. On a network of 3 computers, the computations should only require 4 minutes, with a few seconds for the control program to spread and merge the functions. With a network of 6 computers, the computational time required becomes 2 minutes, but the control program might now take a minute or more. Effective parallel processing demands a strategy for optimizing the advantages.

Machines in parallel process have a variety of heterogeneous designs, ranging from a few powerful processors to thousands of simple individual processing elements. No matter what the architecture, traditional software development methodologies fail to optimize full hardware capabilities. In fact, the efficient use of parallel processors decreases as the number of processors increases. Any increase in performance is limited by the parallel processing strategy; a constraint which presents a major technical barrier to utilizing heterogeneous computing systems for high performance applications.

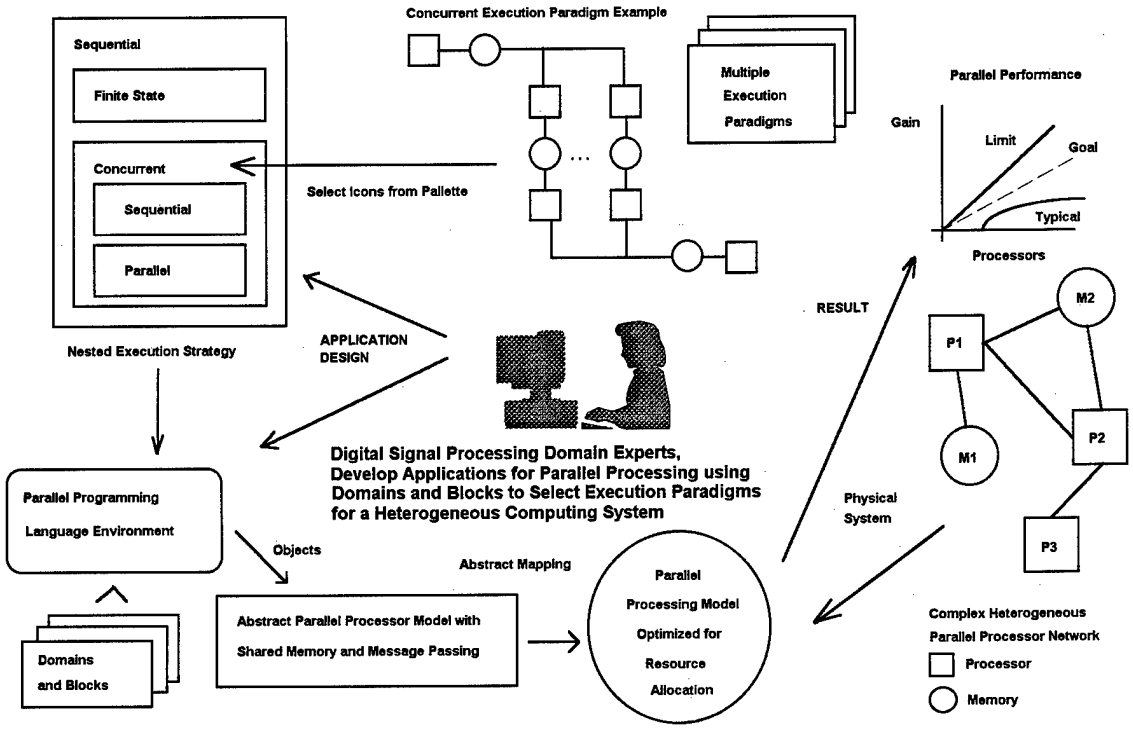
In order to utilize the full parallel processing opportunities of a hardware configuration, SOS uses a unique approach. It creates the parallel execution strategy inside the application through the use of standard

objects, whereas most systems optimize for parallel processing after compilation. The SOS process creates a design using an abstract model that maps to the physical system to execute the resulting application.

This abstract model consists of a number of parallel processors communicating via message passing or shared memory. The single objective is to minimize the overall application execution time. The SOS approach is unique in its optimization step which transforms an application execution paradigm into an efficient parallel processing program for a specific computer architecture.

In order to deal with a large number of diverse parallel processing structures, SOS has chosen to introduce an intermediate abstract architecture based on multiple processors, a shared memory, and message passing for communication as shown in Figure 2.4-1. This architecture is then mapped onto a target parallel processor for execution. These paradigms map an execution strategy into the shared memory model. That model is translated into the target architecture execution strategy and parallel processing begins.

Figure 2.4-1 Parallel Signal Processing with Multiple Execution Paradigms



The user application is mapped onto the target platform and the C++ executable program is generated. The mapping of the abstract model to the target architecture is a resource allocation problem which SOS solved by operations research methods and discrete variable optimization techniques. In doing this optimization, a number of different constraints are considered including the partitioning of the overall execution strategy, the mapping of the execution paradigms onto processors, the scheduling of tasks within a processor, the synchronization of tasks between processors, and the overall data communications and storage. The overall execution time can only be minimized if all these factors are constrained.

The option to specify concurrent processing in a large scale application enables the selection and nesting of execution paradigms. The specific application domains are loaded, which allow the selection of specific blocks to be linked together. A nested map of the execution strategy can be viewed to assess choices for

parallel execution. The parameters for shared memory and message passing can be adjusted to tailor the performance of the execution strategy. Traditionally, parallel processor control operating systems perform load balancing based on ad hoc statistics that recommend each processor to spend the same amount of time on computation and communication. For a specific application this may be the worst computing strategy. In fact, it is the SOS experience that efficient parallel processing always minimizes communication delay and maximizes computation for a given performance period. The SOS approach optimizes the execution of an application, not the performance of the operating system.

2.4.1 Example Software Package Parallel Processing

The best way to understand the SOS phonetic speech recognition approach using parallel processing is to examine an actual application that has real time computing constraints and computationally intensive tasks that would execute on a parallel digital signal processing system. This example is of an SOS phonetic speech recognition system for real time operation on a high performance desktop computer. The speech recognition operates in five major computing phases on the real time speech data collected in the last .01 seconds on a distributed digital signal processor.

The SOS design illustrates an example of an interactive Air Traffic Control training application. The system provides two giga operations and 200 mega flops spread among five processors with nine independent execution paths for interactive processing. This design is a coarse grained parallel desktop computing system for speech recognition applications. The key to achieving the high performance of these systems is software that is able to exploit the parallel processing operations.

Figure 2.4.1-2 Alternative Parallel Processing Design Partitions

Partition By Function

- **Allocate each of the five phases to a processor**
- **Communicate data between phases by shared memory buffers**
- **PRO: Natural division by functions, least programming**
- **CON: Heterogeneous processors with radically different capabilities will make allocation very difficult**

Partition by Data

- **Each processor can do all of the five phase processing**
- **Send data segment to next available processor by shared memory**
- **Communicate results by messages upon completion**
- **PRO: One program works the same on all processors, easy to test**
- **CON: Heterogeneous processors with radically different capabilities will make load balancing very difficult**

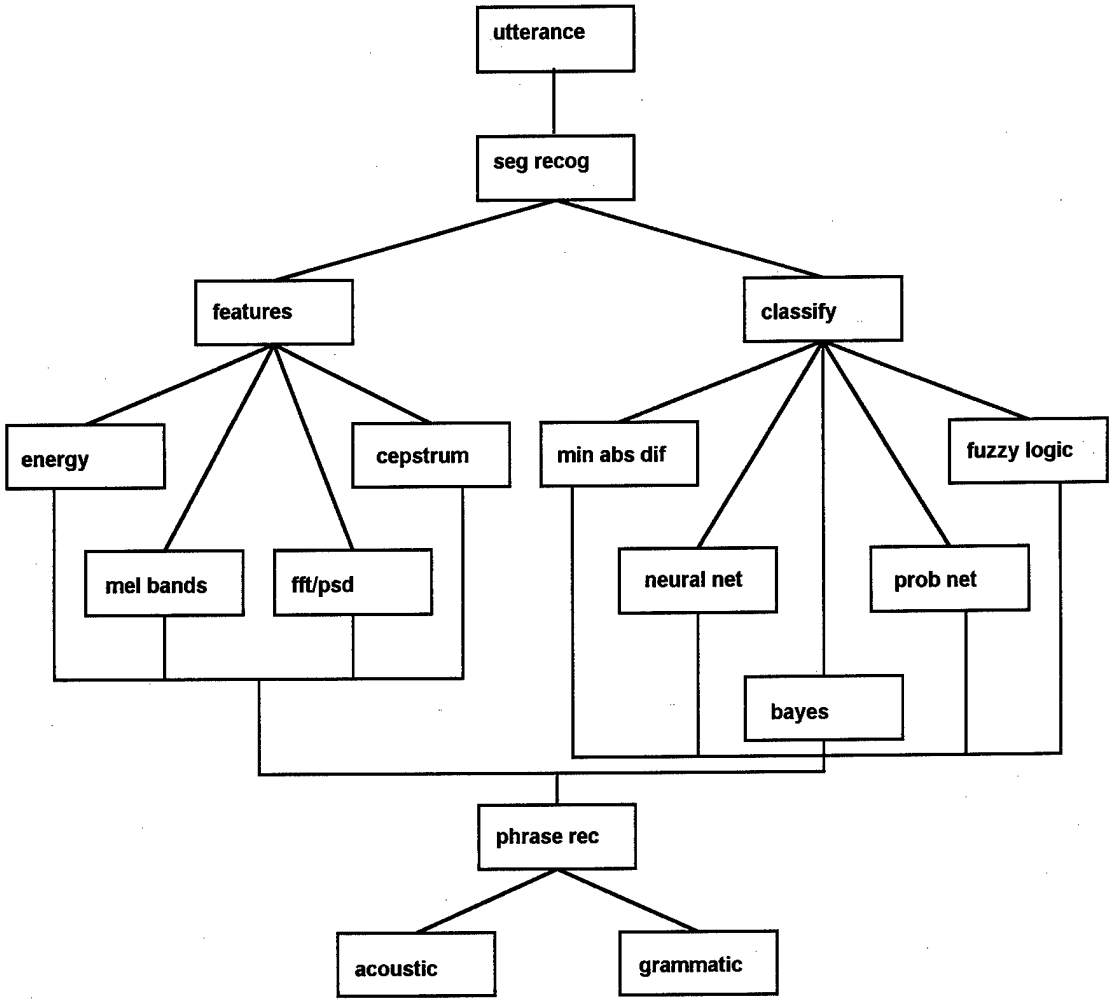
Irregular Partition

- **Pipeline segment data by phases in parallel**
- **Allocate phase to best processor per phase computation**
- **Send data segments by shared memory**
- **Communicate results by messages**
- **PRO: Make best use of processing resources, highest performance**
- **CON: Complex allocation and processor load balancing**

Figure 2.4.1-1 illustrates the real time processing phases of the phonetic speech recognition process. In the utterance and segment phase, the speech data is digitized, filtered, and blocked into segments. In the

features phase, the short time sound segment is processed to determine its phonetic features. In the classify phase, these features are classified to estimate the phoneme using a number of independent parallel algorithms. In the identify phase, the process determines the best estimate for each phonetic segment of the utterance. In the phrase identification phase, the syntactic and semantic processing are performed to output text in a form suitable for word processing, direct interpretation, or further content evaluation. This process is continuous and computationally intensive with ample opportunity for concurrent execution of the multiple recognition phases.

Figure 2.4.1-1 Parallel Data Flow and Computation for Language Module Objects



In a parallel design sense, the phases represent a natural partitioning by function with fixed communication between phases, but with possibly many parallel short time segments. Three different parallel processing allocations can be considered for this heterogeneous system as shown in Figure 2.4.1-2. First, the allocation of functional phases to separate processors with short time segment buffers. Second, the partitioning by short time data segments with the entire process running on each processor. As we strive for the highest performance, it becomes obvious that an irregular partitioning should be considered that can adapt to the changing computational load.

2.4.2 Phonetic Speech Recognition Computation

The speech recognition process uses digital signal processing for phonetic detection and identification of a set of finite length sound units that remain constant for short time periods. These segments are the signals that will be detected and identified in the dynamic and noisy signal processing environment. The steps in this process are for the parallel data flow and signal processing. This is a computationally intensive method that requires high speed digital signal processing to achieve real time performance. The process is scaleable so that more parallel processing resources will lead to more accurate speech recognition. It is also modular so that each processing step can be efficiently distributed over multiple computers for parallel execution.

The parallel signal processing diagram of the phonetic speech recognition is for speech data collected into fixed length blocks for segment processing by the utterance object. The phonetic feature generation step in the recognition phase consists of four parallel processes for each segment of speech. This process can be allocated to DSPs in parallel for any speech time segment as indicated on the parallel blocks.

Figure 2.4.3-1 Parallel Processes for Phonetic Speech Recognition

```
SEQUENTIAL - utterance recognition
        PARALLEL  - segment recognition
                PARALLEL - segment features
                        SEQUENTIAL  - energy
                        PARALLEL - Mel Filters
                                SEQUENTIAL  - FIR filter
                        SEQUENTIAL  - fft/psd
                        SEQUENTIAL  - lpc/cepstrum
                PARALLEL - classification method
                        PARALLEL - minimum absolute difference metric
                                SEQUENTIAL - mad by blocks
                        SEQUENTIAL  - neural network layers
                                PARALLEL - layer nodes
                                        SEQUENTIAL - nodes
                        SEQUENTIAL  - probabilistic neural network layers
                                PARALLEL - pnn layer nodes
                                        SEQUENTIAL - pnn nodes
                        PARALLEL - phoneme set
                                SEQUENTIAL  - fuzzy membership
        SEQUENTIAL  - phrase recognition
                SEQUENTIAL  - phoneme probability estimate
                PARALLEL - phoneme state
                        SEQUENTIAL  - dynamic programming step
                PARALLEL - phonetic groups
                        SEQUENTIAL  - word
        SEQUENTIAL  - lexical phase
                PARALLEL - acoustic generation
                        SEQUENTIAL  - parse tree
                PARALLEL - context group
                        SEQUENTIAL  - context evaluation
```

The phonetic classification step in the recognition stage consists of five parallel processes for each segment of speech. Additional methods may be easily added to this structure for special applications or to explore

new algorithms. Each method operates in parallel and can be distributed onto a set of heterogeneous processors for high speed computing.

The discrete word recognition algorithm processes all of the sound segments in an utterance. These phonemes derived from the utterance are used to perform a sounds like match to the entries in the word dictionary. The word that best matches the utterance is returned with a confidence estimate calculated from the phoneme probabilities.

The final lexical processing uses a modified parsing algorithm that operates on two levels for acoustics and grammatics. First the acoustic level using the phonetic word data produces multiple candidate phrases for an utterance. Second, the grammar level selects the most likely phrase based on non acoustic language information. This process stands on a context free grammar which defines the phrase structure for the language. This grammar is applied in a top down approach to select the candidate phrase that best fits the language. Semantic tests are applied to reject nonsensical phrases, create contractions, format numbers, etc. and punctuate where possible.

2.4.3 Parallel Processing for Phonetic Speech Recognition

The design of the processes for phonetic speech recognition are limited to the two choices of sequential and parallel for simplicity of presentation in Figure 2.4.3-1. Sequential indicates that the following processes indented one level are required to be executed in order, not in parallel. Parallel indicates that the following processes indented one level are allowed to be executed in parallel, but all must be complete to advance to the next process. As a result, the inner most process is always sequential as would be expected, but significant opportunity exists for concurrent computation. The strategy for this concurrent computation depends on the physical architecture.

Figure 2.4.3-2 Allocation of Computing Tasks to Processors

PROCESSOR	PHASE 1	PHASE 2	PHASE 3	PHASE 4	PHASE 5
P54c - A		energy	fuzzy logic	phon sel	grammatic
P54c - B		cepstrum	bayes	prob est	acoustic
320C40 - C	filter	fir 1-8	neural network		
320C40 - D	filter	fir 9-18	prob neural net		
320C80 - E		fft/psd	min abs diff	dyn prog	

The computation for the time consuming segment recognition process is allocated among the five physical processors in the architecture as shown in Figure 2.4.3-2. The five phases of the computation are the columns and the processors are rows. Each entry shows the computation allocation for that phase to that processor. This allocation optimizes the major portion of the recognition cycle.

Figure 2.4.3-3 Segment Processing Timeline

Segment	.01 sec	.01 sec	.01 sec
1	digitize data	Phase 1, 2, 3, 4, 5	
2		digitize data	Phase 1, 2, 3, 4, 5
3			digitize data

The overall execution timeline is illustrated in Figure 2.4.3-3. This shows the overlapping computation possible for each recognition segment where data collection occupies .01 seconds which can be overlapped on each of the five recognition phases in a segment. Based on the physical architecture this allocates 200

Mips or 20 megaflops to the segment computation. Using multiple execution paradigms, alternative allocations could be used to experiment with overall system performance.

2.4.4 Summary of the Phonetic Speech Recognition Example

This example illustrates the range and power of the SOS phonetic speech recognition concept. The processing diagrams are selected from digital signal processing domains and composed to solve a complex real time speech recognition problem with multiple parallel processors. Figure 2.4.4-1 lists the benefits of object oriented programming for this example. The traditional development of this type of application would involve numerous programmers, DSP experts, and system designers. It would require a long expensive development; trying alternative architectures for the distribution of the computations would not be economically feasible. The SOS approach avoids this difficulty with its multiple execution paradigms and object oriented programming.

Figure 2.4.4-1 Benefits of Object Oriented Programming in this Example

Top down processing architecture definition and modular functional design
Multiple distributed processing allocation evaluation to select best computing design
Modular allocation of computing to parallel processors to optimize performance
Real time interrupts and rapid processing response time in a high level language
Design and programming combined in a single environment for rapid development
Extensive reuse of application domains and blocks for cost effective implementation
Interactive testing to verify computing requirements of the application
Validation of real time performance by C++ code generation for target architecture
Intuitive graphical documentation for maintenance and modification

2.5 Software Package Hardware Specification

The selection of commercial off the shelf hardware for the different aspects of the Phase II project concerns the effective mix and match of both analog and digital equipment. The analog components are the speech input microphone, audio amplifier, low pass filters, analog to digital converters, and any noise rejection units. The digital components are the analog to digital converter, sound input digital signal processing, general purpose computing, network interface units, and parallel digital signal processors.

The general SOS engineering guidelines for selecting and modifying hardware are to use components with standard interfaces such as signal levels and buses, to use components with proven performance and reliability as documented by manufacturers and users, and to select components that can be upgraded with a minimum impact on the existing Software Package design and implementation. In all cases, SOS will resist the prototype engineering temptation to design custom hardware, to develop non standard interfaces, or to modify existing commercial equipment. The use of unmodified off the shelf commercial components and standard interfaces is the key to commercializing a prototype product in a timely manner, to provide ease of manufacturing for predictable production costs, and for the often rapid inventory expansion of a new product.

The hardware used will develop, test, and integrate both the hardware and software for the operational Navy ATC facility training configurations and the commercial Software Package product. Initially, SOS will use its existing Pentium PCs networked together along with the DSP development systems for the TI320C50 and the TI320C30. These units will be used to interface to sound cards for PC compatibility,

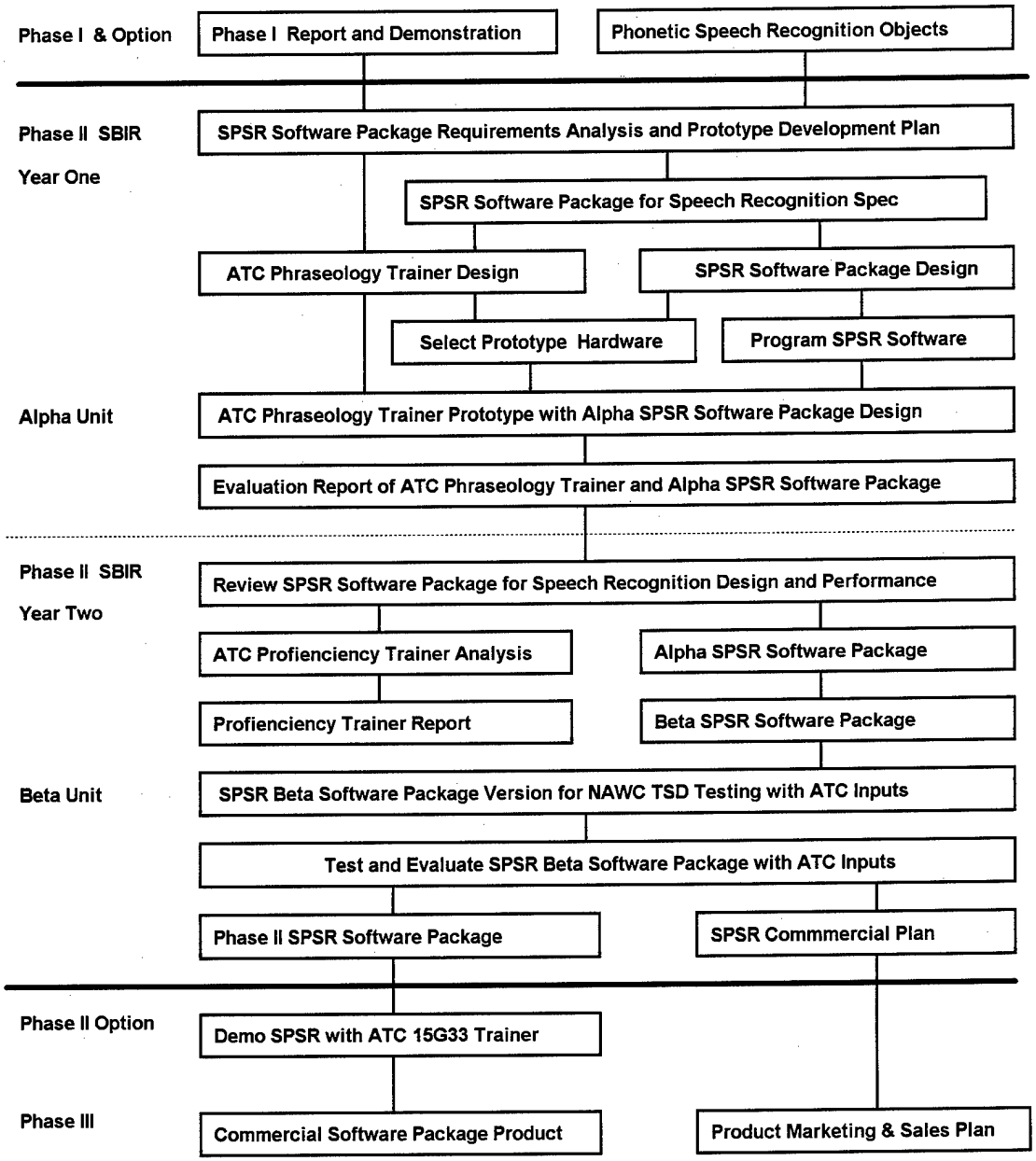
phone line interfaces for telephone compatibility, and line level interfaces for audio compatibility. SOS has existing multi track recording and audio test equipment to support sample speech collection and recorded testing.

It is expected that this development network will be upgraded with the addition of a 200 Mhz Pentium Pro unit with MMX similar to the one SOS designed for IMS to support its speech compression and recognition research. These units run the standard Windows 95, Windows NT, and UNIX/LINUX/POSIX operating systems and support C++ object oriented software development. As required, SOS will add a TI320C80 development system for multimedia interface and development. In general the DSP software will be developed with the SPOX operating system. SOS has access to SUN, HP, SGI, and other workstations for network interface testing but, does not intend to develop software on these platforms. The exact hardware selection for the alpha and beta product configuration will be made at the commercial product design review in Phase III of the project. Whether the software package commercial configuration is hardware inclusive is a decision to be determined after the market research is performed near the end of Phase II. Changes will be initiated depending on cost, availability, compatibility, and performance.

3.0 PHASE II DEVELOPMENT PLAN

The SPSR software development plan for Phase II takes a two pronged approach. One side is the conceptualization, construction, and testing of the SOS Standard Objects for Phonetic Speech Recognition software package, which will be ongoing for the duration of the Phase II. Progress will begin from the point where the Phase I demonstration ended. The capabilities of the software to detect and categorize phonemes will be routinely and sequentially increased while the introduction of the International Phonetic Alphabet Dictionary of 180,000 entries will be implemented.

Figure 3.0-1 Phase II Work Plan for the Software Package for Speech Recognition



The second focus will be to develop the prototype deliverables mentioned in Section 1.4. The concept for both these items emerged from Phase I research, and their development is an efficient and natural offshoot of the phonetic speech recognition research. The deliverables are the stand alone Phraseology Trainer (alpha unit) to be delivered and evaluated by the end of the first year in Phase II. The performance of this prototype will be expanded and the features enhanced for development as a stand alone SPSR tool kit (beta unit) ready for testing and evaluation before the end of year two in the Phase II project with NAWC TSD supplied ATC speech input. Figure 3.0.1 summarizes the work plan during Phase II which is described in more detail in the following sections.

3.1 Phase II Task Definitions

A summary of the Phase II tasks for the software package is presented in Figure 3.1-1. The first task is a complete requirements analysis for the software package, as it will drive the Naval ATC training architecture and the alpha and beta prototype units which will follow. The accomplishment of each task level described in the figure will refine the software package as it progresses through development. A software package definition and design will then be written in the format of a commercial product specification to maximize the commercial potential of market acceptance and product recognition.

As part of the product specification and development, the software package will be performance tested in the Phraseology Trainer alpha configuration and analyzed for use in the 15G33 Proficiency Trainer for operational interface evaluation in a beta configuration. Multiple iterations on the design of the software package component are anticipated until a stable product is available for delivery to the Navy ATC facility during year two for independent testing and evaluation by actual users in the Navy ATC operational environment. A final software package product specification for use in the Phase III commercialization will be generated which reflects the Phase II research results, the prototype development process, and the operational user feedback.

Figure 3.1-1 PHASE II WORK PLAN TASKS AND DOCUMENTATION

TASK	DOCUMENTATION
YEAR ONE	
1 Requirements Analysis	Requirements Definitions for Deliverables
2 Software Package Design	SPSR Software Package Specification
3 Alpha Prototype Definition	Phraseology Trainer Demonstration
4 Alpha Prototype Delivery	SPSR Software Walk Through Design Review
5 Prototype Evaluate & Implement	Phase II SBIR Interim Report
YEAR TWO	
6 Performance Analysis & Tuning	SPSR Software Package & Alpha Progress
7 Beta Prototype Definition	15G33 Proficiency Trainer Analysis
8 Beta Prototype Delivery	SPSR Performance and Interface Report
9 Beta Operational Evaluation	SPSR ATC Test and Analysis Report
10 Final Software Package	Phase II SBIR Final Report
PHASE II OPTION	
11 Demo of 15G33 and SPSR	Proficiency Trainer Using SPSR Report

3.1.1 Requirements Analysis Task

The requirements analysis task will define and collect requirements for the software package and the alpha and the beta deliverables. The information will be gathered from existing published documents, by interviewing Navy personnel, and by applying SOS phonetic speech recognition engineering experience. The requirements will be divided into four classes consisting of functional, performance, user, and interface. Functional requirements define the logical and physical characteristics of the components. Performance requirements define the component behaviors and define the limitations of the system responses. User requirements define the operational states and modes from the operators viewpoint. Interface requirements define the external and internal data objects that exist in the software package and their physical and logical communication. These requirements will be reviewed formally by Navy personnel to assure that the resulting deliverables will satisfy the intended ATC training applications.

3.1.2 Software Package Development Task

The SPSR software package development task will create the following six software objects. A top level control object directs the computations of the lower level objects. The lower level objects divide into two groups for speech recognition and system interface. The speech recognition group contains the phonetic speech recognition objects, the multiple execution paradigm objects, and the lexical processing objects. The interface group contains the objects for human computer interaction and local area network interaction. The software will be developed primarily in C++ using the SOS object oriented approach, however, it may be necessary to utilize assembly code for small portions of the DSP software.

3.1.3 Alpha Prototype Development Task, Year 1

The SOS alpha prototype development task is to deliver to the Navy ATC facility a demonstration Phraseology Trainer using the SPSR software in year one. The design will be based on the Phase I demonstration model with changes to reflect information collected in the requirements analysis task. The off the shelf hardware selection for the alpha prototype will be made at the end of the requirements review. SOS expects to use a Pentium class microprocessor with an add on DSP board and interfaced to sound cards for PC compatibility with head mounted line level microphones. The SPSR speech recognition software package and a top level control software will be integrated with the hardware to demonstrate a Phraseology Trainer prototype performance in a stand alone mode.

3.1.4 Alpha Performance Analysis Task, Year 1

The SPSR performance analysis task will include operational Navy ATC command and control environment usage with various student scenarios and noise conditions. The analysis will focus on satisfaction of functional, performance, and user requirements. The performance analysis task will form the basis of evaluation of the effectiveness of the speech recognition research alpha prototype. The emphasis in the performance analysis task will be on qualitative and quantitative test design and data collection.

3.1.5 Alpha Prototype Tuning Task, Year 1

The tuning task uses the alpha SPSR prototype and the results of the performance analysis task to refine the recognition parameters and optimize the overall performance of the Phraseology Trainer. User

evaluations augment performance data to provide feedback for improving and tuning the system for continued use and evaluation during years one and two.

3.1.6 Beta Prototype Development Task, Year 2

The beta SPSR prototype development task is to deliver a fully integrated package for ongoing advanced testing during year two using NAWC TSD spoken ATC data. The beta prototype will evidence much of the technology demonstrated in the alpha prototype, but will have features specifically designed for continuing ATC training derived from the requirements analysis performed at the start of the project. The beta prototype will require the more advanced iterations of the software package which will be in continuing development during year two and will utilize off the shelf hardware to be determined by a market review.

3.1.7 Beta Performance Analysis Task, Year 2

The beta performance analysis task is generated from the operational evaluations received by users of the Proficiency Trainer as they relate to the requirements task. As the beta prototype is placed into service the operators provide feedback on the operational efficiency and satisfaction of the functional and user requirements. The performance analysis task stipulates alterations and improvements necessary to optimize performance of the SPSR as part of the operational Navy ATC training scenario. The success criteria defined in the requirements analysis task will form the basis of an overall measure of effectiveness. The emphasis will be on quantified measures of performance that can be used in a Phase III commercialization effort to develop a successful and marketable product.

3.1.8 Beta Prototype Tuning Task, Year 2

The beta prototype tuning task takes the data from the SPSR performance analysis task and makes the changes required to advance the performance of the software. It is anticipated that most alterations will involve clarification of instructions and fortifying the system against "crash attacks" from uneducated users. The tuning task also pertains to the software package, which will be integrated into Naval ATC training architecture during year two.

3.2 Software Package Specification Document

A primary portion of the documentation for the SBIR Phase II research and development project is an SPSR Software Package Specification. This specification will define the exact product configuration of the software package as delivered to the Navy ATC training facility. The process for developing a technical specification requires attention to detail. It is an iterative process beginning after the completion of the requirements review in the first months of the Phase II project. The critical software package elements are defined by the design review and developed for configuration in the Phraseology Trainer prototype. The alpha SPSR configuration and stand alone test results are used for additional tuning, and the subsequent specification is the principle portion of the interim Phase II first year report in month twelve. In the second year of this research project, the beta SPSR prototype is integrated and tested at the Navy ATC facility with a primary emphasis on tuning performance and updating the software package specification to reflect the results of operational experience. The SOS Standard Objects for Phonetic Speech Recognition Specification will provide manufacturers specifications for configurations, compatibility and performance of the completed Version 1.0 of the software package, as developed for and tested by the Naval ATC training facility.

3.3 Phase II Final Report and Option Task

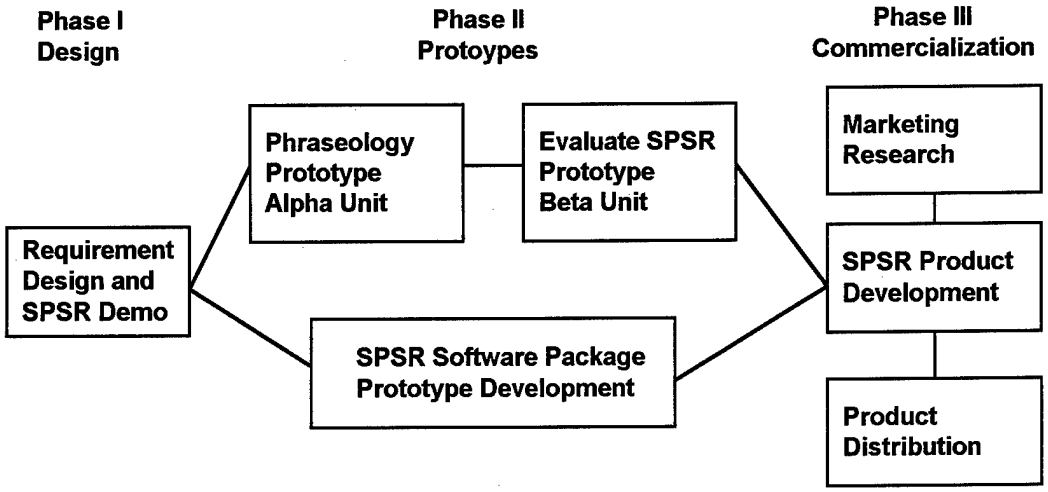
The Phase II final report preparation task will concentrate on a complete documentation of the work performed in Phase II, an analysis of the performance of the deliverables, and the plans for Phase III commercialization of the of the SPSR software package and the Phraseology Trainer using the SOS Standard Objects for Phonetic Speech Recognition Software Package. Two deliveries are anticipated for the Phase II report: an interim report at the end of year one, and a final report and the end of year two. The interim report delivered half way through the research effort will concentrate on the requirements and the stand alone prototype design, performance analysis and ongoing tuning of the Phraseology Trainer. The final report will provide detail description of the stand alone SPSR beta prototype design, implementation and performance tuning in year two. The SPSR software package research will be reviewed, and the developmental stages will be outlined, concentrating on the evaluation of the integrated prototype as a commercial product. The Phase II report will be delivered at the end of the twenty four month project effort. The Phase II Option Report will describe the demonstration of the SPSR software with the Navy 15G33 ATC Proficiency Trainer.

4.0 PHASE III COMMERCIALIZATION PLAN

The research for this project was designed to take a two pronged approach, and the same logic applies to commercialization. At the end of the Phase II effort it is anticipated that the SOS Standard Objects for Phonetic Speech Recognition software package will be ready for commercialization as an off the shelf speech recognition component to be used in a myriad of application domains. Our market research indicates an increasing world wide demand for object oriented speech recognition systems, for phonetic speech recognition environments, and for parallel DSP programming for complex speech recognition applications on desktop and networked systems.

The second emphasis of the commercialization plan is the sale to commercial entities of Phraseology and Proficiency Trainers. Whereas the customer base for the prototype trainers may appear limited, they will serve as demonstrative models for customers requiring customized products which are based on the technology presented in the two training versions. During Phase II professional assistance will be contracted for the development and future implementation of a formal marketing plan emphasizing management planning, communications, and sales presentations for funding with complete production and promotion budgets. Figure 4.0-1 summarizes the commercialization outline.

Figure 4.0-1 SPSR Software Package for Speech Recognition Product Plan



4.1 Software Package Commercialization

The success of the Phase II research will be used to present the SOS Standard Objects for Phonetic Speech Recognition as a commercial software product which will be developed and integrated for a parallel DSP environment as illustrated in the market opportunity matrix in Figure 4.1.1. A prototype concept evaluation version of the product will be tested internally and used to create a market test model that will be distributed to a limited number of potential customers. The final product will be integrated into an SOS product line and sold to the custom speech recognition tools market after proper planning for efficient and reliable production and for economical and rapid distribution. The objectives for strategic marketing will involve competitive product positioning, market definition, identification of competition, manufacturing and production, sales and distribution, and an analysis of funding.

Figure 4.1-1 Market Opportunity Matrix

OPPORTUNITY	COMMERCIALIZATION	CUSTOMERS
1 Phonetic Speech Recognition	1 Speech Recognition Tools	1 IBM, BBN, SRI,
2 Parallel DSP Programming	2 Object Software Libraries	2 Apple, SCO, IBM
3 Speech Application Systems	3 Microprocessor Systems	3 SUN, DEC, HP, SGI
4 Voice Network Environments	4 Custom Applications	4 AT&T, TI, Motorola
	5 OEM Speech Products	

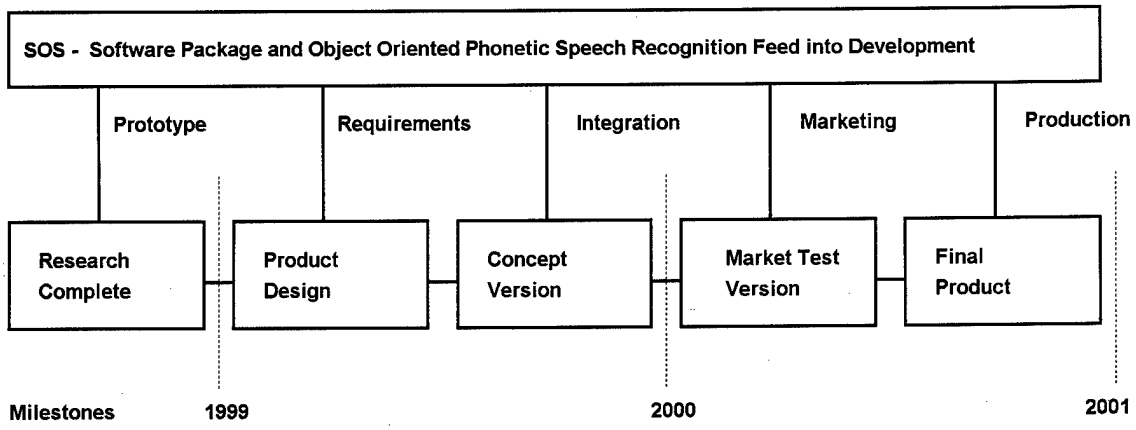
4.1.1 Production Considerations

Production parameters will be determined based upon the commercial specification prepared in Phase II for the SOS Standard Objects for Phonetic Recognition software package. To this will be applied the requirements for interface for microphone and other off the shelf applications with which the software must be compatible. Supply sources, manufacture facilities, production methods, quality control capabilities, and ownership interests will be negotiated with assistance of professional consultants based upon the marketing plan developed prior to the completion of the Phase II.

4.1.2 Distribution Considerations

A distribution strategy will be determined after the marketing plan defines the target market, its size, the SOS anticipated share of the market and the growth potential of the market. Retail versus licensing options will be examined. An anticipated approach is to become a value added component of an existing product line with established distribution channels. The inclusion of the SOS Standard Objects for Phonetic Speech Recognition into computer games is an example of this market directive. At this point in the current technology environment, the windows of opportunity open and close with increasing rapidity, often with a click on the Internet. Industrial partners will be interviewed during the entire SBIR Phase II process, with the understanding that distribution implies not just the ability to move product, but the ability to advertise, market and promote it as well.

Figure 4.1-2 Commericializaton Path and Milestones



The commercialization path is shown in Figure 4.1-2 with key milestones and anticipated dates. It begins with the completion of Phase II research and development for the software package which results in a prototype product at the end of 1999. SOS will conduct market research into the demand and product definition to create application domains for the commercial market. These product requirements will feed

into the product design in 2000 which will result in a commercial product. Customer feedback and market analysis will be used to define this product for a post year 2000 production.

4.2 Commercialization to Air Traffic Control Entities

The Phraseology and Proficiency Trainers are products which were designed for a single user, the Naval ACT training facility. It is a clear and obvious objective to market these prototype products to other air traffic control training entities. The customer base in this country would be limited to the FAA, the Air Force, and the colleges which present air traffic control as a field of study. Marketing the equipment to training facilities in foreign countries opens up a much larger customer base, and is made possible by the multi-lingual feature of the SOS Standard Objects for Phonetic Speech Recognition technology which is based on an international phonetic alphabet with a capability of approximately 350 languages.

4.3 Commercialization to Non Air Traffic Control Entities

The Phraseology Trainer has applications anywhere students learn to speak a new or unusual vernacular. One example is police personnel being trained for dispatch duties. This requires strict usage of code phrases which require practice. The Phraseology Trainer can easily be customized for such a scenario, and the existing prototype version serves well as a demonstration model. One example is use by Emergency 911 operators. Personnel requiring advanced continuing professional education could hone their skills and, in a practice context, confront emergency situations beyond the magnitude of any tragedy (hopefully) they confront in their everyday work setting.

4.4 Applications of Speech Recognition

A large number of speech recognition applications have been identified in the published literature. The following set of specific applications are from direct proposals by SOS to specific customers and devices investigated by SOS. Estimates for the expanding market size of speech recognition equipment exceed two billion dollars in the next few years.

4.4.1 Computer Aided Transcription

The computer aided transcription (CAT) industry supplies products for medical and legal transcription. Court stenographers enter phonetic representations of speech at over 240 words per minutes using a chord keyboard. These phonetic entries are transcribed to text with over 99% accuracy. SOS has proposed the conversion of the IPA output of the phonetic speech recognition process to the keystroke entries of stenographers to automate this process. In this application, speakers will be identified and the recognized phonemes will be converted to stenographic keystrokes that are compatible with transcription systems.

4.4.2 Voice Controlled Video Camera

A number of voice operated devices for hands free and eyes free operation have been built. SOS has designed and prototyped a voice controlled video surveillance camera control system. The product was developed using the Standard Objects for Phonetic Speech Recognition in C++. It is designed to provide a low cost implementation in commercial security systems where it will allow a single roving guard to monitor multiple cameras through an audio/video link. The system is designed to support up to sixteen cameras along with automated digitized change detection and anomaly warning to reduce false alarm calls. The system is designed for use by security equipment system manufacturers to reduce the cost of large area patrol environments.

4.4.3 Automatic Speaker Identification

The use of automatic speaker identification for security and access control has been implemented in a small number of locations. SOS has proposed a unique application for phonetic speech recognition to the Navy for communication control. Government furnished syntax from standard Naval communication protocols will be analyzed to create a hierarchical system for communication priority control. The recognized speech will be speaker independent from a defined vocabulary of combat tactical information. The utterances will be used to synthesize, localize, classify, prioritize, alert, and catalog mission essential communications.

The goal is to increase the confidence and reliability of the battle group for shared tactical communications. The watch stations communication priority control intelligent voice recognition system will provide a speaker independent capability to process verbal information using the commands and syntax encountered in a tactical situation to accomplish information transfer that includes automatic prioritization, classification, alerts, and mission cataloging. The requirements of this application will be used to create an application design that can use the SOS phonetic speech recognition objects.

4.4.4 Spoken Accent Correction

A commercial implementation of large scale accent correction was implemented by NTT the Japanese Telephone Company. English students in Japan can call a voice recognition system and repeat an English phrase which is evaluated for the correct accent and pronunciation. At a recent test in Japan, native English speakers were graded over 90% correct except for one individual with a strong regional accent. There is a demand for a product that would perform intelligent accent evaluation and remedial correction at the phoneme/word basis. SOS has explored this as a potential product for the language teaching industry.

4.4.5 Spoken Language Translation

The problem of spoken language translation has been addressed by simple pocket voice synthesizers for travelers and complex multiple language conferencing as is used by the United Nations. Translation is a complex problem and SOS has addressed two applications for virtual reality communication in proposals to the Air Force and the Army.

Interface applications for phonetic speech recognition at the Armstrong Laboratory include a number of existing audio projects. An example is the Integrated Audio Technology Demonstrator which currently integrates several new audio technologies. The 3-D Audio Display System can also be interfaced to the phonetic speech recognition to provide virtual reality speech perception based on content recognition such as names or locations. Telerobotic systems provide a natural speech interface for hands free and eyes free operational environments that would be lethal to human workers. These interfaces can be achieved by the application of the Standard Objects for Phonetic Speech Recognition software. This innovative technology provides a robust speech recognition interface that does not require training, but can adapt to specific users for efficient operation.

For the Army, SOS proposed to develop a specific speech recognition application to allow OPFOR personnel to communicate efficiently with a simulated environment for contingency planning training. The prototype contingency planning scenario would be based on a finite state model that describes all of the possible scenario states and the transitions into and from each state. A set of rules would control the use of resources and the effects of input actions.

4.5 Competition in the Speech Recognition Marketplace

The competition in the speech recognition marketplace ranges from very large organizations such as AT&T to small start up research companies like SOS. The market is expanding and the amount of competition is a sign of the healthy expansion of the industry. Figure 4.5-1 is a partial list of the major competitors in the speech recognition industry that exhibited at the 1996 Advanced Speech Applications & Technologies ASAT conference in San Francisco.

Figure 4.5-1 Competitors in the Speech Recognition Marketplace

ORGANIZATION	PRODUCT
AcuVoice	AV1200 Speech synthesis
Apple Computer Inc.	PlainTalk, Chinese Dictation Kit
ALTECH	MIT Technology for automating telephone services
AT&T	WATSON Advanced Speech Application Platform
BBN	Hark speech recognizer
Berkely Speech Technologies	BeSTspeech text to speech translator
Dialogic	Antares speech recognition
Dragon Systems	DragonDictate speech recognition
Entropic	TrueTalk speech recognition
First Byte	Speech synthesis Pro Voice
France Telecom	CNET research and development
Gentex	Noise canceling microphones
IBM	VoiceType Dictation system
Lernhout & Hauspie	Multi lingual speech recognition
Linkon	Maestro speech processing
Motorola	Lexicus natural speech interface
Nuance	Speech recognition software
Philips	SpeechMagic continuous speech recognition
PureSpeech	Continuous speaker independent recognition
Speach Interface	Portable industrial speech recognition
Speech Systems	Phonetic Engine speech recognition
Novell	SRAPI consrotium standard for speech recognition
TI	TI Multiserve speech telecommunications
T-NETIX	SpeakEZ voice identification
Unisys	NL assistant speech recognition
Verbex	VoiceBrowser for Net Scape Navigator
Voice Control	Telecommunication speaker verification
Voice Processing	Vpro automated speech recognition

4.6 Patents and Intellectual Property Protection

The field of speech recognition has been exploited by inventors for patent protection since the 1930s. Numerous computer based speech recognition programs have been funded by ARPA and other government organizations since the 1950s each usually resulting in new patent applications. The principal investigator and others have filed for patent protection of this unique method of phonetic speech recognition as an improvement to existing patents listed in the references. In addition, SOS has copyrighted its Standard Objects for Phonetic Speech Recognition and maintains its multiple execution paradigm methodology as a trade secret. This level of intellectual property protection has generally been sufficient for a new product development in this field.

5.0 Conclusion

In conclusion, the SOS Phase I and Option period research project was successful, as documented in this SBIR Phase I report. SOS is prepared in Phase II to design a language based SPSR speech recognition software package; to develop a fully functioning system using Standard Objects for Phonetic Speech Recognition; to create application interfaces that allow users to evaluate the package in complex applications on multiple computing platforms and networks; and to implement that user feedback. During Phase I, an outline of the commercialization plan for a language based SPSR speech recognition product was developed. It will be expanded during Phase II based on operational experience to prepare for a Phase III product development. The anticipated SOS Phase III product is a multi-platform signal processing program and execution environment targeted toward the commercial and industrial development of real time and embedded speech recognition applications. The primary customers for this product are in the speech recognition application market.

This research and development effort will be directed in Phase II to the development and testing of a fully functioning prototype Software Package, and in Phase III to the production of a commercial speech recognition product. The Software Package design incorporates the SOS Standard Objects for Phonetic Speech Recognition system that uses the international phonetic alphabet (IPA) as the smallest identified spoken unit. It supports both discrete word recognition and continuous speaker independent recognition with automatic training for a multi lingual speaker environment.

In addition, two prototype training systems were defined in Phase I that will be developed in Phase II. These systems are an ATC Phraseology Trainer and an ATC Proficiency Trainer. These prototype training systems will act as a test bed for the Software Package in Phase II. A proof of concept prototype of the Phraseology Trainer was demonstrated in Phase I for speaker independent continuous speech recognition.

The applications for reliable speech recognition as a computer input mechanism are limitless. The microphone is the next human machine interface paradigm; a touch pad and headset will eliminate the need for a traditional keyboard. This is a technology with intense international competition, one that presents a market advantage for the economy of the country that creates a usable product with a successful proprietary technology. SOS is fully prepared to develop a Software Package for Speaker Independent or Dependent Speech Recognition Using Standard Objects for Phonetic Speech Recognition. We are confident that the Phase II project will be completed successfully, within budget, and on schedule. The resulting Phase III commercial product will exceed the performance specifications to become an integral component of computers in the twenty first century.

REFERENCES

[1] "From Text to Speech: The MITalk System," Allen, et al, Cambridge University Press, 1987.
Discussion of Lexical processing, phonetic basis of speech synthesis, and alphabets.

[2] "Readings in Speech Recognition," Alex Waibel & Kai-Fu Lee, Morgan Kaufmann Publishers, 1990.
Digital signal processing applications to speech signals, homomorphic filters, etc.

[3] "Discrete Time Processing of Speech Signals," J. Deller, J. Proakis, & J. Hansen,MacMillian, 1993.
Short time Fourier analysis of speech signals and spectrum estimates.

[4] "Advanced Algorithms and Architectures for Speech Understanding," G. Pirani, Springer, 1985.
Esprit Study concentrates on the use of the Mel scale for speech recognition.

[5] "Finding Groups in Data," L. Kaufmann & P. Rousseeuw, Wiley, 1990.
Metric methods for grouping and identifying objects by features statistically.

[6] "Signal and Image Processing with Neural Networks," T. Masters, Wiley, 1994.
Multilayer feed forward deterministic neural network signal processing methods.

[7] "Classification Algorithms," Mike James, Wiley, 1985.
Bayesian and statistical classification and identification methods.

[8] "Advanced Algorithms for Neural Networks," T. Masters, Wiley, 1995.
Probabilistic neural network models for classification applications.

[9] "Neural Networks and Fuzzy Systems," Bart Kosko, Prentice Hall, 1992.
Fuzzy logic and sets for classification and identification under uncertainty.

[10] TIMIT CD-ROM, U.S. Department of Commerce, 1991.
Corpus of five hours of phoneme labeled american speech with accents, ages, sex, etc.

[11] "Phonetic Speech Recognition for Japanese Hiragana," Henry L. Pfister, Fuzzy Logic 95, 1995.

[12] "State Recognition for Noisy Dynamic Systems," Henry L. Pfister, Technology 2005, 1995.

[13] "Some Computer Organizations and their Effectiveness," IEEE Transactions on Computers, Vol C-21 No 9, pp 948, Sept, 1972.

[14] "User's Guide to the P4 Parallel Programming System," Ralph Butler and Ewing Lusk, Argonne National Laboratory, 1995.

[15] "DOME: Parallel Programming in a Heterogeneous Multi-User Environment," Adam Beguelin et al. Carnegie Mellon University, Supercomputing '95, April, 1995.

[16] "Portable Programs for Parallel Processors," Lusk, Overbeek, et al. Holt, Rinehart, and Winston, 1987.

[17] "Breaking Moore's Law," John Wharton, Microprocessor Report, Published by Micro Design Resources, Volume 9, Number 6, May, 1995, pp 15. ISSN 0899-9341

[18] "Your Next Mainframe," Andy Reinhart, BYTE, May, 1995, pp 48.

[19] "An Introduction to Parallel Programming," K. Chandy and S. Taylor, Jones and Bartlett Publishers, Boston, 1992.

[20] "Parallel Programming with PCN," I. Foster and S. Tuecke, Argonne National Laboratory, 1993.

[21] "Visual Programming with Multiple Execution Paradigms," Technical Research Report 95-01, Standard Object Systems, Inc., 1995.

[22] "Visual Programming of Parallel Processors," Technical Research Report 95-02, Standard Object Systems, Inc., 1995.

[23] "Exploring Parallel Processing," Edward Rietman, Windcrest Publishers, 1990.

[24] "Real Time Computer Speech Recognition System," T. A. Bordeaux, U.S. Patent 4,852,170, 1989.

[25] "Voice Recognition Method by Analyzing Syllables," A. Tanaka, U. S. Patent 5,129,000, 1992.

[26] "Speech Recognition Apparatus Capable of Discriminating between Similar Acoustic Features of Speech," A. Amano, N. Hataoka, S. Yajima, I. Musashino, U. S. Patent 4,998,280, 1991.

[27] "Method for Interactive Speech Recognition and Training," J. Roberts, J. Baker, E. Porter, U. S. Patent 5,027,406, 1991.

[29] "Speech Recognition System for Nautral Language Translation," P. Brown, A. Pietra, J. Pietra, F. Jelinek, R. Mercer, U. S. Patent 5,293,584, 1994.

[30] "System and Method for Facilitating Speech Transcription," H. Pfister, G. Smith, M. Tsuchiya, U. S. Patent Application 08/278,266, Filed 1994.

[31] "Experiences Using MODSIM and C++," H. L. Pfister, Intelligence Community Modeling and Simulation Symposium, Washington D.C., 1992.

[32] "Object Oriented Planning," H. L. Pfister, 2nd Aerospace Conference on AI, Los Angeles, 1988.

[33] "Object Oriented Simulation," H. L. Pfister, First Aerospace Conference on AI, Los Angeles, 1987.

[34] "Object Oriented Design and Programming in C++," H. L. Pfister, THITI, Bankok, Thailand, 1991.

[35] "Speaker Dependent Speech Compression for Low Bandwidth Communication," Henry Pfister, 1996 IEEE Aerospace Applications Conference, Snowmass, 1996.

GLOSSARY

ARPABET	Advanced Research Project Agency phonetic alphabet
CAT	Computer Aided Transcription
CODEC	Coder Decoder for telephone analog to digital and digital to analog data
DARPA	Defense Advanced Research Project Agency now ARPA
DSP	Digital Signal Processor
FAM	Fuzzy Associative Memory
FFT	Fast Fourier Transform
FIR	Finite Impulse Response digital filter
HMM	Hidden Markov Model
IPA	International Phonetic Alphabet
LPC	Linear prediction Coding
MCM	Markov Chain Model
MLFN	Multi Layer Feedforward Network
PSD	Power Spectral Density
SOS	Standard Object Systems, Inc.
TIMIT	Texas Instruments and Massachusetts Institute of Technology
OEM	Original Equipment Manufacturer